An Interop Publication

CONNEXIONS®

The Interoperability Report

March 1995 | *Special Issue: NetWorld+Interop 95 Las Vegas Companion* | Volume 9, No. 3

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, a division of SOFTBANK Exposition and Conference Company, 303 Vintage Park Drive, Foster City, California, 94404–1138, USA.
Phone: +1 (415) 578-6900
Fax: +1 (415) 525-0194
E-mail: connexions@interop.com

Subscription hotline: 1-800-575-5717
or +1-502-493-3217

ISSN 0894–5926

From the Editor

Welcome to Las Vegas and to *NetWorld+Interop 95*. This expanded edition of *ConneXions* is being provided to all attendees while supplies last. It contains several articles related to the conference and tutorial program. At the end of each article you will find pointers to specific conference events where you can hear more about the technologies being discussed in this publication. *ConneXions* is a monthly technical journal covering all aspects of computer networking and interoperability. It is the only official companion publication to the conference, and has been published by Interop Company since the first Interop conference in 1987. For an index of our 96 back issues, send e-mail to: `connexions@interop.com`.

The Internet has been growing exponentially for several years and continues to outgrow even the most optimistic estimates. (See the results of the latest Internet Domain Survey on page 47). One aspect of this growth is a depletion of the available 32-bit address space. The Internet Engineering Task Force (IETF) has been working on this problem for a number of years, and has recently made a recommendation for a new version of the Internet Protocol (IP) based on a 128-bit addressing scheme. This "IP Next Generation" (IPng) is described in our first article by Bob Hinden. Tim Dixon follows the IPng article with a look at how IPng might affect the OSI effort.

Fibre Channel is a fusion of networking and channel approaches to device and system interconnection. Peter Walford and Ed Frymoyer describe this emerging high-speed technology. You will also be able to see examples of Fibre Channel systems on the exhibition floor.

"Network Management" is almost synonymous with SNMP, the *Simple Network Management Protocol*. Originally developed by a small group of dedicated Internet engineers, SNMP has seen wide implementation deployment both inside and outside the traditional networking industry. SNMP++ is not a new version of SNMP, but rather a tool for network management programming. Kim Banker and Peter Mellquist give an overview of SNMP++.

Videoconferencing is becoming a more common way to conduct business meetings. Traditionally, videoconferencing systems use special-purpose hardware and software and dedicated high-speed data lines, all of which come at a considerable cost. But it is possible to design low-cost systems using a combination of desktop computers, inexpensive cameras and the Internet as your backbone. Tim Dorcey describes the popular CU-SeeMe videoconferencing system.

You will find more information on all of these topics in our extensive conference and tutorial program, and we hope you will continue to receive updates with a subscription to *ConneXions*.

IP Next Generation Overview

by Robert M. Hinden, Ipsilon Networks

Introduction

This article presents an overview of the *Next Generation Internet Protocol* which was recommended by the IPng Area Directors of the *Internet Engineering Task Force* (IETF) at the Toronto IETF meeting on July 25, 1994, and documented in RFC 1752, "The Recommendation for the IP Next Generation Protocol" [16]. The recommendation was approved by the Internet Engineering Steering Group on November 17, 1994 and made a Proposed Standard.

The formal name of this protocol is IPv6 (the "6" refers to it being assigned version number 6). The current version of the Internet Protocol is version 4 ("IPv4"). This article is intended to give the reader an overview of the IPng protocol. For more detailed information the reader should consult the documents listed in the reference section.

IPng is a new version of IP which is designed to be an evolutionary step from IPv4. It is a natural increment to IPv4. It can be installed as a normal software upgrade in internet devices and is interoperable with the current IPv4. Its deployment strategy was designed to not have any "flag" days. IPng is designed to run well on high performance networks (e.g., ATM) and at the same time is still efficient for low bandwidth networks (e.g., wireless). In addition, it provides a platform for new internet functionality that will be required in the near future.

This article describes the work of IETF IPng working group. Several individuals deserve specific recognition. These include Steve Deering, Paul Francis, Bob Gilligan, Dave Crocker, Ran Atkinson, Jim Bound, Ross Callon, Bill Fink, Ramesh Govindan, Christian Huitema, Erik Nordmark, Tony Li, Dave Katz, Yakov Rekhter, Bill Simpson, and Sue Thompson.

Key issues for IPng

There are several key issues that should be considered when reviewing the design of the next generation internet protocol. Some are very straightforward. For example the new protocol must be able to support large global internetworks. Others are less obvious. There must be a clear way to transition the current large installed base of IPv4 systems. It doesn't matter how good a new protocol is if there isn't a practical way to transition the current operational systems running IPv4 to the new protocol.

Growth

Growth is the basic issue which caused there to be a need for a next generation IP. If anything is to be learned from our experience with IPv4 it is that the addressing and routing must be capable of handling reasonable scenarios of future growth. It is important that we have an understanding of the past growth and where the future growth will come from.

Currently IPv4 serves what could be called the "computer market." The computer market has been the driver of the growth of the Internet. It comprises the current Internet and countless other smaller internets which are not connected to the Internet. Its focus is to connect computers together in the large business, government, and university education markets. This market has been growing at an exponential rate. One measure of this is that the number of networks in current Internet (40,073 as of 10/4/94) is doubling approximately every 12 months. The computers which are used at the endpoints of Internet communications range from PCs to Supercomputers. Most are attached to Local Area Networks (LANs) and the vast majority are not mobile.

New markets

The next phase of growth will probably not be driven by the computer market. While the computer market will continue to grow at significant rates due to expansion into other areas such as schools (elementary through high school) and small businesses, it is doubtful it will continue to grow at an exponential rate. What is likely to happen is that other kinds of markets will develop. These markets will fall into several areas. They all have the characteristic that they are extremely large. They also bring with them a new set of requirements which were not as evident in the early stages of IPv4 deployment. The new markets are also likely to happen in parallel with one another. It may turn out that we will look back on the last ten years of Internet growth as the time when the Internet was small and only doubling every year. The challenge for an IPng is to provide a solution which solves today's problems and is attractive in these emerging markets.

Nomadic personal computing devices seem certain to become ubiquitous as their prices drop and their capabilities increase. A key capability is that they will be networked. Unlike the majority of today's networked computers they will support a variety of types of network attachments. When disconnected they will use RF wireless networks, when used in networked facilities they will use infrared attachment, and when docked they will use physical wires. This makes them an ideal candidate for internetworking technology as they will need a common protocol which can work over a variety of physical networks. These types of devices will become consumer devices and will replace the current generation of cellular phones, pagers, and personal digital assistants. In addition to the obvious requirement of an internet protocol which can support large scale routing and addressing, they will require an internet protocol which imposes a low overhead and supports auto configuration and mobility as a basic element. The nature of nomadic computing requires an internet protocol to have built in authentication and confidentiality. It also goes without saying that these devices will need to communicate with the current generation of computers. The requirement for low overhead comes from the wireless media. Unlike LANs which will be very high speed, the wireless media will be several orders of magnitude slower due to constraints on available frequencies, spectrum allocation, error rates, and power consumption.

Another market is networked entertainment. The first signs of this emerging market are the proposals being discussed for 500 channels of television, video on demand, etc. This is clearly a consumer market. The possibility is that every television set will become an Internet host. As the world of digital high definition television approaches, the differences between a computer and a television will diminish. As in the previous market, this market will require an internet protocol which supports large scale routing and addressing, and auto configuration. This market also requires a protocol suite which imposes the minimum overhead to get the job done. Cost will be the major factor in the selection of an appropriate technology.

Another market which could use the next generation IP is device control. This consists of the control of everyday devices such as lighting equipment, heating and cooling equipment, motors, and other types of equipment which are currently controlled via analog switches and in aggregate consume considerable amounts of electrical power. The size of this market is enormous and requires solutions which are simple, robust, easy to use, and very low cost. The potential pay-back is that networked control of devices will result in cost savings which are extremely large.

continued on next page

IP Next Generation Overview *(continued)*

The challenge the IETF faced in the selection of an IPng was to pick a protocol which meets today's requirements and also matches the requirements of these emerging markets. These markets will happen with or without an IETF IPng. If the IETF IPng is a good match for these new markets it is likely to be used. If not, these markets will develop something else. They will not wait for an IETF solution. If this should happen it is probable that because of the size and scale of the new markets the IETF protocol would be supplanted. If the IETF IPng is not appropriate for use in these markets, it is also probable that they will each develop their own protocols, perhaps proprietary. These new protocols would not interoperate with each other. The opportunity for the IETF was to select an IPng which has a reasonable chance to be used in these emerging markets. This would have the very desirable outcome of creating an immense, interoperable, world-wide information infrastructure created with open protocols. The alternative is a world of disjoint networks with protocols controlled by individual vendors.

Transition

At some point in the next three to seven years the Internet will require a deployed new version of the Internet protocol. Two factors are driving this: *routing* and *addressing*. Global internet routing based on the on 32-bit addresses of IPv4 is becoming increasingly strained. IPv4 addresses do not provide enough flexibility to construct efficient hierarchies which can be aggregated. The deployment of *Classless Inter-Domain Routing* (CIDR) [6] is extending the lifetime of IPv4 routing by a number of years, the effort to manage the routing will continue to increase. Even if the IPv4 routing can be scaled to support a full IPv4 Internet, the Internet will eventually run out of network numbers. There is no question that an IPng is needed, but only a question of when.

The challenge for an IPng is for its transition to be complete before IPv4 routing and addressing break. The transition will be much easier if IPv4 address are still globally unique. The two transition requirements which are the most important are flexibility of deployment and the ability for IPv4 hosts to communicate with IPng hosts. There will be IPng-only hosts, just as there will be IPv4-only hosts. The capability must exist for IPng-only hosts to communicate with IPv4-only hosts globally while IPv4 addresses are globally unique.

The deployment strategy for an IPng must be as flexible as possible. The Internet is too large for any kind of controlled rollout to be successful. The importance of flexibility in an IPng and the need for interoperability between IPv4 and IPng was well stated in a message to the SIPP mailing list by Bill Fink, who is responsible for a portion of NASA's operational internet. In his message he said:

> "Being a network manager and thereby representing the interests of a significant number of users, from my perspective it's safe to say that the transition and interoperation aspects of any IPng is *the* key first element, without which any other significant advantages won't be able to be integrated into the user's network environment. I also don't think it wise to think of the transition as just a painful phase we'll have to endure en route to a pure IPng environment, since the transition/coexistence period undoubtedly will last at least a decade and may very well continue for the entire lifetime of IPng, until it's replaced with IPngng and a new transition. I might wish it was otherwise but I fear they are facts of life given the immense installed base.

> "Given this situation, and the reality that it won't be feasible to coordinate all the infrastructure changes even at the national and regional levels, it is imperative that the transition capabilities support the ability to deploy the IPng in the piecemeal fashion... with no requirement to need to coordinate local changes with other changes elsewhere in the Internet...
>
> "I realize that support for the transition and coexistence capabilities may be a major part of the IPng effort and may cause some headaches for the designers and developers, but I think it is a duty that can't be shirked and the necessary price that must be paid to provide as seamless an environment as possible to the end user and his basic network services such as e-mail, FTP, Gopher, X-Window clients, etc...
>
> "The bottom line for me is that we must have interoperability during the extended transition period for the base IPv4 functionality..."

Another way to think about the requirement for compatibility with IPv4 is to look at other product areas. In the product world, backwards compatibility is very important. Vendors who do not provide backward compatibility for their customers usually find they do not have many customers left. For example, chip makers put considerable effort into making sure that new versions of their processor always run all of the software that ran on the previous model. It is unlikely that Intel would develop a new processor in the X86 family that did not run DOS and the tens of thousands of applications which run on the current versions of X86s.

Operating system vendors go to great lengths to make sure new versions of their operating systems are binary compatible with their old version. For example the labels on most PC or Mac software usually indicate that they require OS version XX or greater. It would be foolish for Microsoft come out with a new version of Windows which did not run the applications which ran on the previous version. Microsoft even provides the ability for Windows applications to run on their new OS NT. They understand that it was very important to make sure that the applications which run on Windows also run on NT.

The same requirement is also true for IPng. The Internet has a large installed base. Features need to be designed into an IPng to make the transition as easy as possible. As with processors and operating systems, it must be backwards compatible with IPv4. Other protocols have tried to replace TCP/IP, for example XTP and OSI. One element in their failure to reach widespread acceptance was that neither had any transition strategy other than running in parallel (sometimes called "Dual Stack"). New features alone are not adequate to motivate users to deploy new protocols. IPng must have a great transition strategy and new features.

History of the IPng effort

The IPng protocol represents the evolution of many different IETF proposals and working groups focused on developing an IPng. It represents over three years of effort focused on this topic. A brief history follows:

By the Winter of 1992 the Internet community had developed four separate proposals for IPng. These were "CNAT," "IP Encaps," "Nimrod," and "Simple CLNP." By December 1992 three more proposals followed; "The P Internet Protocol" (PIP), "The Simple Internet Protocol" (SIP) and "TP/IX."

continued on next page

IP Next Generation Overview *(continued)*

In the Spring of 1992 the "Simple CLNP" evolved into "TCP and UDP with Bigger Addresses" (TUBA) and "IP Encaps" evolved into "IP Address Encapsulation" (IPAE).

By the fall of 1993, IPAE merged with SIP while still maintaining the name SIP. This group later merged with PIP and the resulting working group called themselves "Simple Internet Protocol Plus" (SIPP). At about the same time the TP/IX Working Group changed its name to "Common Architecture for Next Generation Internet Protocol" (CATNIP). [25]

The IPng area directors made a recommendation for an IPng in July of 1994. This recommendation, from [16], includes the following elements:

- Current address assignment policies are adequate.
- There is no current need to reclaim underutilized assigned network numbers.
- There is no current need to renumber major portions of the Internet.
- CIDR-style assignments of parts of unassigned Class A address space should be considered.
- "Simple Internet Protocol Plus (SIPP) Spec. (128 bit ver)" [19] be adopted as the basis for IPng.
- The documents listed in Appendix C be the foundation of the IPng effort.
- An IPng Working Group be formed, chaired by Steve Deering and Ross Callon.
- Robert Hinden be the document editor for the IPng effort.
- An IPng Reviewer be appointed and that Dave Clark be the reviewer.
- An Address Autoconfiguration Working Group be formed, chaired by Dave Katz and Sue Thomson.
- An IPng Transition Working Group be formed, chaired by Bob Gilligan and TBA.
- The Transition and Coexistence Including Testing Working Group be chartered.
- Recommendations about the use of non-IPng addresses in IPng environments and IPng addresses in non-IPng environments be developed.
- The IESG commission a review of all IETF standards documents for IPng implications.
- The IESG task current IETF working groups to take IPng into account.
- The IESG charter new working groups where needed to revise old standards documents.
- Informational RFCs be solicited or developed describing a few specific IPng APIs.
- The IPng Area and Area Directorate continue until main documents are offered as Proposed Standards in late 1994.

- Support for the Authentication Header be required.
- Support for a specific authentication algorithm be required.
- Support for the Privacy Header be required.
- Support for a specific privacy algorithm be required.
- An "IPng framework for firewalls" be developed.

IPng overview

IPng is a new version of the Internet Protocol, designed as a successor to IP version 4 [15]. IPng is assigned IP version number 6 and is formally called IPv6 [14].

IPng was designed to take an evolutionary step from IPv4. It was not a design goal to take a radical step away from IPv4. Functions which work in IPv4 were kept in IPng. Functions which didn't work were removed. The changes from IPv4 to IPng fall primarily into the following categories:

- *Expanded Routing and Addressing Capabilities:* IPng increases the IP address size from 32 bits to 128 bits, to support more levels of addressing hierarchy and a much greater number of addressable nodes, and simpler auto-configuration of addresses. The scalability of multicast routing is improved by adding a "scope" field to multicast addresses. A new type of address called a "cluster address" is defined, to identify topological regions rather than individual nodes. The use of cluster addresses in the IPng source route allows nodes to control the path through which their traffic flows.

- *Header Format Simplification:* Some IPv4 header fields have been dropped or made optional, to reduce the common-case processing cost of packet handling and to keep the bandwidth cost of the IPng header as low as possible despite the increased size of the addresses. Even though the IPng addresses are four times longer than the IPv4 addresses, the IPng header is only twice the size of the IPv4 header.

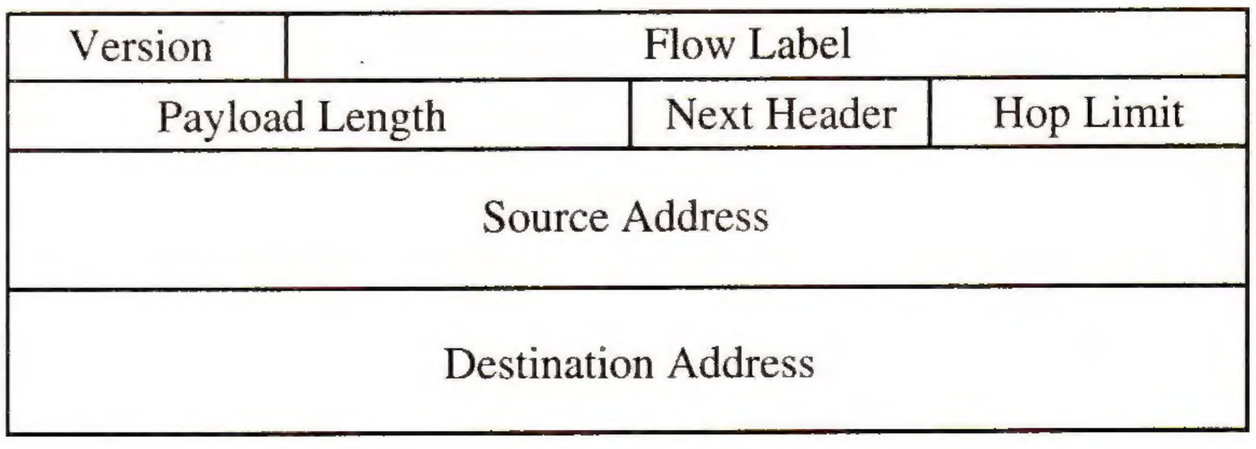
- *Improved Support for Options:* Changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.

- *Quality-of-Service Capabilities:* A new capability is added to enable the labeling of packets belonging to particular traffic "flows" for which the sender requests special handling, such as non-default quality of service or "real-time" service.

- *Authentication and Privacy Capabilities:* IPng includes the definition of extensions which provide support for authentication, data integrity, and (optionally) confidentiality. This is included as a basic element of IPng and will be included in all implementations.

Header format

The IPng protocol consists of two parts, the basic *header* and *options.*



continued on next page

IP Next Generation Overview *(continued)*

The fields have the following meanings:

Version: 4-bit Internet Protocol version number = 6.

Flow Label: 28-bit field. See "Quality-of-service capabilities" section.

Payload Length: 16-bit unsigned integer. Length of payload, i.e., the rest of the packet following the IPng header, in octets.

Next Header: 8-bit selector. Identifies the type of header immediately following the IPng header. Uses the same values as the IPv4 Protocol field [21].

Hop Limit: 8-bit unsigned integer. Decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero.

Source Address: 128 bits. An address of the initial sender of the packet. See [1] for details.

Destination Address: 128 bits. An address of the intended recipient of the packet (possibly not the ultimate recipient, if an optional Routing Header is present).

Options

IPng includes an improved option mechanism over IPv4. IPng options are placed in separate headers that are located between the IPng header and the transport-layer header in a packet. Most IPng option headers are not examined or processed by any router along a packet's delivery path until it arrives at its final destination. This facilitates a major improvement in router performance for packets containing options. In IPv4 the presence of any options requires the router to examine all options. The other improvement is that unlike IPv4, IPng options can be of arbitrary length and the total amount of options carried in a packet is not limited to 40 bytes. This feature plus the manner in which they are processed, permits IPng options to be used for functions which were not practical in IPv4. A good example of this is the IPng Authentication and Security Encapsulation options.

In order to improve the performance when handling subsequent option headers and the transport protocol which follows, IPng options are always an integer multiple of 8 octets long, in order to retain this alignment for subsequent headers.

The IPng option headers which are currently defined are:

Option	Function
Routing:	Extended Routing (like IPv4 loose source route).
Fragmentation:	Fragmentation and Reassembly.
Authentication:	Integrity and Authentication.
Security Encapsulation:	Confidentiality.
Hop-by-Hop Option:	Special options which require hop-by-hop processing.
End-to-End Options:	Optional information to be examined by the destination node.

Addressing

IPng addresses are 128 bits long and are identifiers for individual nodes and sets of nodes. IPng addresses refer to a single node interface. Each interface may be assigned multiple IPng addresses. There are three types of IPng addresses. These are *unicast, cluster,* and *multicast.* Unicast addresses identify a single node. Cluster addresses identify a group of nodes that share a common address prefix, such that a packet sent to a cluster address will be delivered to one member of the group.

Multicast addresses identify a group of nodes, such that a packet sent to a multicast address is delivered to all of the nodes in the group. There are no broadcast addresses in IPng, their function being superseded by multicast addresses.

IPng supports addresses with four times the number of bits compared to IPv4 addresses (128 versus 32). This is 4 Billion times 4 Billion (2^{96}) times the size of the IPv4 address space (2^{32}). This works out to be:

340,282,366,920,938,463,463,374,607,431,768,211,456

This is an extremely large address space. In a theoretical sense this is approximately 665,570,793,348,866,943,898,599 addresses per square meter of the surface of the planet Earth (assuming the Earth surface is 511,263,971,197,990 square meters).

In more practical terms the assignment and routing of addresses requires the creation of hierarchies which reduces the efficiency of the usage of the address space. Christian Huitema performed an analysis in [12] which evaluated the efficiency of other addressing architectures (including the French telephone system, USA telephone systems, current Internet using IPv4, and IEEE 802 nodes). He concluded that 128-bit IPng addresses could accommodate between $8x10^{17}$ to $2x10^{33}$ nodes assuming efficiency in the same ranges as the other addressing architectures. Even using his most pessimistic estimate this would provide 1,564 for addresses per square meter of the surface of the planet Earth. The optimistic estimate would allow for 3,911,873,538,269,506,102 addresses per square meter of the surface of the planet.

Format Prefix

The specific type of IPng address is indicated by the leading bits in the address. The variable-length field comprising these leading bits is called the *Format Prefix* (FP). The initial allocation of these prefixes is as follows:

Allocation	*Prefix (binary)*	*Fraction of Address Space*
Reserved	0000 0000	1/256
Reserved	0000 0001	1/256
NSAP Allocation	0000 001	1/128
IPX Allocation	0000 010	1/128
Reserved	0000 011	1/128
Reserved	0000 100	1/128
Reserved	0000 101	1/128
Reserved	0000 110	1/128
Reserved	0000 111	1/128
Reserved	0001	1/16
Reserved	001	1/8
Provider-Based Unicast Adr	010	1/8
Reserved	011	1/8
Reserved for Geographic Adr	100	1/8
Reserved	101	1/8
Reserved	110	1/8
Reserved	1110	1/16
Reserved	1111 0	1/32
Reserved	1111 10	1/64
Reserved	1111 110	1/128
Reserved	1111 1110 0	1/512
Link Local Use Addresses	1111 1110 10	1/1024
Site Local Use Addresses	1111 1110 11	1/1024
Multicast Addresses	1111 1111	1/256

continued on next page

IP Next Generation Overview *(continued)*

This supports the direct allocation of provider addresses, NSAP addresses, IPX addresses, local use addresses, and multicast addresses. Space is reserved for geographic addresses. The remainder of the address space is reserved for future use. This can be used for expansion of existing use (e.g., additional provider addresses, IPX addresses, etc.) or new uses (e.g., separate locators and EID).

Fifteen percent of the address space is initially allocated. The remaining 85% is reserved for future use.

Unicast addresses

There are several forms of unicast address assignment in IPng. These are global provider hierarchical unicast addresses, geographical hierarchical addresses, NSAP hierarchical addresses, IPX hierarchical addresses, local-use addresses, and IP-only host addresses. Additional addresses types can be defined in the future. The assignment plan for unicast addresses is described in [3] and [2].

Provider-based unicast addresses

Provider-based unicast addresses are used for global communication. They are similar in function to IPv4 addresses under CIDR. Their format is:

3	n bits	m bits	p bits	125–n–m–p
010	PROVIDER ID	SUBSCRIBER ID	SUBNET ID	NODE ID

The first 3 bits identify the address as a provider-oriented address. A provider ID is assigned to Internet service providers, which then assign portions of the address space to subscribers. This usage is similar to assignment of IP addresses under CIDR. The SUBSCRIBER ID distinguishes among multiple subscribers attached to the provider identified by the PROVIDER ID. The SUBNET ID identifies a topologically connected group of nodes within the subscriber network identified by the subscriber prefix. The NODE ID identifies a single node among the group of nodes identified by the subnet prefix.

Local-use address

A local-use address is a unicast address that has only local routability scope (within the subnet or within a subscriber network), and may have local or global uniqueness scope. They are intended for use inside of a site for "plug and play" local communication and for bootstrapping up to a single global addresses.

There are two types of local-use unicast addresses defined. These are *Link-Local* and *Site-Local*. The Link-Local-Use is for use on a single link and the Site-Local-Use is for use in a single site. Link-Local-Use addresses have the following format:

10 bits	n bits	p bits
1111111010	0	NODE ID

Link-Local-Use addresses are designed to be used for addressing on a single link for purposes such as auto-address configuration.

Site-Local-Use addresses have the following format:

10 bits	n bits	m bits	p bits
1111111011	0	SUBNET ID	NODE ID

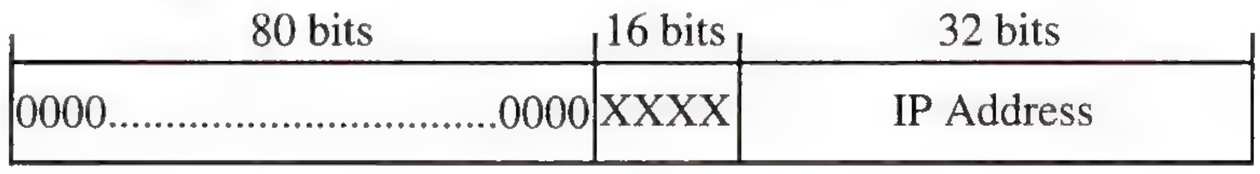
For both types of local use addresses the NODE ID is an identifier which must be unique in the domain in which it is being used. In most cases these will use a node's IEEE-802 48-bit address. The SUBNET ID identifies a specific subnet in a site.

The combination of the SUBNET ID and the NODE ID to form a local use address allows a large private internet to be constructed without any other address allocation.

Local-use addresses allow organizations that are not (yet) connected to the global Internet to operate without the need to request an address prefix from the global Internet address space. Local-use addresses can be used instead. If the organization later connects to the global Internet, it can use its SUBNET ID and NODE ID in combination with a global prefix (e.g., PROVIDER ID+SUBSCRIBER ID) to create a global address.

IPv4-only addresses

IPng unicast addresses are assigned to IPv4-only hosts as part of the SIT scheme for transition from IPv4 to IPng [20]. Such addresses have the following form:



The high-order 80 bits of the address identify the address as an IPv4 address. Bits 80–95 distinguish between an IPv4-only node and an IPng node.

Cluster addresses

Cluster addresses are unicast addresses that are used to reach the "nearest" one (according to unicast routing's notion of nearest) of the set of boundary routers of a cluster of nodes identified by a common prefix in the IPng unicast routing hierarchy. These are used to identify a set of nodes. The cluster address, when used as part of a route sequence, permits a node to select which of several providers it wants to carry its traffic. A cluster address can only be used as a destination address. This capability is sometimes called "source selected policies." Cluster addresses have the general form:

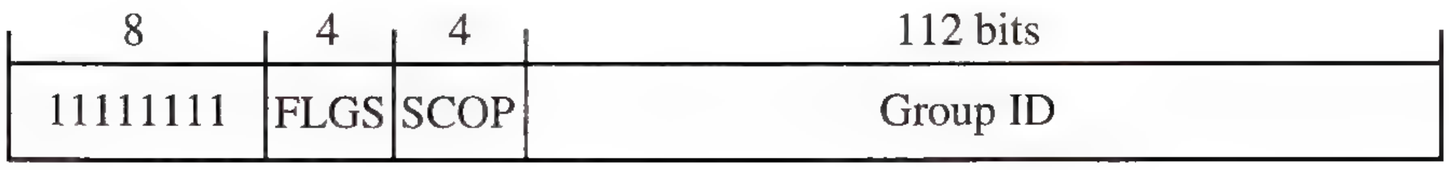
The cluster identifier is an assigned identifier which is unique within the scope of the ADDRESS PREFIX.

Cluster boundary routers are required to be configured to know that they are boundary routers and with the appropriate CLUSTER IDENTIFIER(S). A boundary router may have one or more cluster addresses assigned to it and is required to accept datagrams addressed to the corresponding cluster address as being addressed to itself.

Cluster addresses are most commonly used as intermediate addresses in an IPng Routing Header, to cause a datagram to be routed via one or more specific clusters on the way to its final destination. Cluster addresses must not be used as source addresses in any IPng datagrams.

Multicast addresses

An IPng multicast address is an identifier for a group of nodes. A node may belong to any number of multicast groups. Multicast addresses have the following format:



continued on next page

IP Next Generation Overview *(continued)*

The fields have the following meanings:

11111111 at the start of the address identifies the address as being a multicast address.

FLGS is a set of 4 flags:

0	0	0	T

The high-order 3 flags are reserved, and must be initialized to 0.

T = 0 indicates a permanently-assigned ("well-known") multicast address, assigned by the global Internet Numbering Authority.

T = 1 indicates a non-permanently-assigned ("transient") multicast address.

SCOP is a 4-bit multicast scope value used to limit the scope of the multicast group. The values are:

0	reserved	8	organization-local scope
1	node-local scope	9	(unassigned)
2	link-local scope	A	(unassigned)
3	(unassigned)	B	community-local scope
4	(unassigned)	C	(unassigned)
5	site-local scope	D	(unassigned)
6	(unassigned)	E	global scope
7	(unassigned)	F	reserved

GROUP ID identifies the multicast group, either permanent or transient, within the given scope.

Routing

Routing in IPng is almost identical to IPv4 routing under CIDR, except that the addresses are 128-bit IPng addresses instead of 32-bit IPv4 addresses. With very straightforward extensions, all of IPv4's routing algorithms (OSPF, RIP, IDRP, IS–IS, etc.) can be used to route IPng.

IPng also includes simple routing extensions which support powerful new routing functionality. These capabilities include:

- Provider Selection (based on policy, performance, cost, etc.)
- Host Mobility (route to current location)
- Auto-Readdressing (route to new address)

The new routing functionality is obtained by creating sequences of IPng addresses using the IPng Routing option. The Routing option is used by an IPng source to list one or more intermediate nodes (or topological clusters) to be "visited" on the way to a packet's destination. This function is very similar to IPv4's Loose Source and Record Route option.

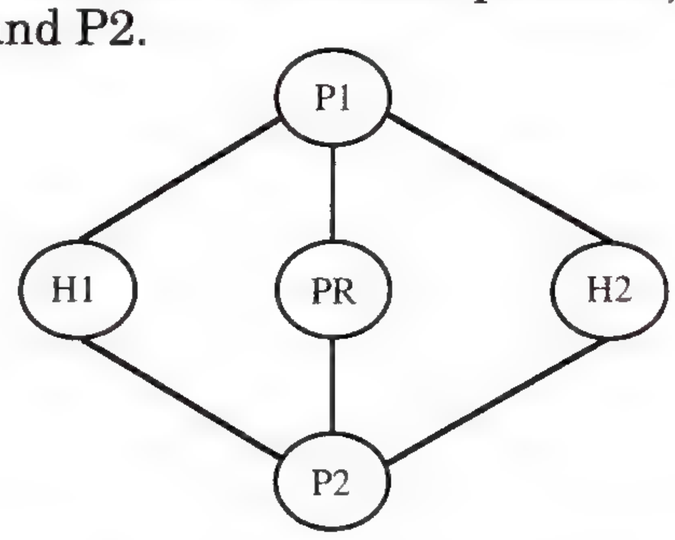
In order to make address sequences a general function, IPng hosts are required to reverse routes in a packet it receives containing address sequences, in order to return the packet to its originator. This approach is taken to make IPng host implementations from the start support the handling and reversal of source routes. This is the key for allowing them to work with hosts which implement the new features such as provider selection or extended addresses.

Three examples show how the address sequences can be used. In these examples, address sequences are shown by a list of individual addresses separated by commas. For example:

SRC, I1, I2, I3, DST

Where the first address is the source address, the last address is the destination address, and the middle addresses are intermediate addresses.

For these examples assume that two hosts, H1 and H2 wish to communicate. Assume that H1 and H2's sites are both connected to providers P1 and P2. A third wireless provider, PR, is connected to both providers P1 and P2.



The simplest case (no use of address sequences) is when H1 wants to send a packet to H2 containing the addresses:

H1, H2

When H2 replied it would reverse the addresses and construct a packet containing the addresses:

H2, H1

In this example either provider could be used, and H1 and H2 would not be able to select which provider traffic would be sent to and received from.

If H1 decides that it wants to enforce a policy that all communication to/from H2 can only use provider P1, it would construct a packet containing the address sequence:

H1, P1, H2

This ensures that when H2 replies to H1, it will reverse the route and the reply would also travel over P1. The addresses in H2's reply would look like:

H2, P1, H1

If H1 became mobile and moved to provider PR, it could maintain (not breaking any transport connections) communication with H2, by sending packets that contain the address sequence:

H1, PR, P1, H2

This would ensure that when H2 replied it would enforce H1's policy of exclusive use of provider P1 and send the packet to H1's new location on provider PR. The reversed address sequence would be:

H2, P1, PR, H1

The address sequence facility of IPng can be used for provider selection, mobility, and readdressing. It is a simple but powerful capability.

Quality-of-service capabilities

The *Flow Label* field in the IPng header may be used by a host to label those packets for which it requests special handling by IPng routers, such as non-default quality of service or "real-time" service. This labeling is important in order to support applications which require some degree of consistent throughput, delay, and/or jitter.

continued on next page

IP Next Generation Overview *(continued)*

The Flow Label is a 28-bit field, internally structured into two subfields as follows:

A flow is a sequence of packets sent from a particular source to a particular (unicast or multicast) destination for which the source desires special handling by the intervening routers. The nature of that special handling might be conveyed to the routers by a control protocol, such as a resource reservation protocol, or by information within the flow's packets themselves, e.g., in a hop-by-hop option.

There may be multiple active flows from a source to a destination, as well as traffic that is not associated with any flow. A flow is identified by the combination of a Source Address and a non-zero FLOW ID. Packets that do not belong to a flow carry a FLOW ID of zero.

A FLOW ID is assigned to a flow by the flow's source node. New FLOW IDs must be chosen (pseudo-)randomly and uniformly from the range 1 to FFFFFF hex. The purpose of the random allocation is to make any set of bits within the FLOW ID suitable for use as a hash key by the routers, for looking up the special-handling state associated with the flow. A FLOW ID must not be re-used by a source for a new flow while any state associated with the previous usage still exists in any router.

The TCLASS subfield provides a means separate from the FLOW ID for a source to identify the desired delivery priority of its packets, relative to other packets from the same source. The TCLASS values are divided into two ranges: values 0 through 7 are used to label flow-controlled packets, e.g., packets that belong to a TCP connection, and values 8 through 15 are used to label non-flow-controlled packets, e.g., "real-time" packets being sent without any flow-control feedback from the receivers.

For flow-controlled traffic, the following TCLASS values are recommended for particular application categories:

0: uncharacterized traffic
1: "filler" traffic (e.g., netnews)
2: unattended data transfer (e.g., e-mail)
3: (reserved)
4: attended bulk transfer (e.g., FTP, NFS)
5: (reserved)
6: interactive traffic (e.g., telnet, X)
7: internet control traffic (e.g., routing protocols, SNMP)

For non-flow-controlled traffic, the lowest TCLASS value (8) should be used for those packets that the sender is most willing to have discarded under conditions of congestion (e.g., high-fidelity video traffic), and the highest value (15) should be used for those packets that the sender is least willing to have discarded (e.g., low-fidelity audio traffic). There is no relative ordering implied between the flow-controlled classes and the non-flow-controlled classes.

Security

The current Internet has a number of security problems and lacks effective privacy and authentication mechanisms below the application layer. IPng remedies these shortcomings by having two integrated options that provide security services. These two options may be used singly or together to provide differing levels of security to different users. This is very important because different user communities have different security needs.

The first mechanism, called the "IPng Authentication Header," is an option which provides authentication and integrity (without confidentiality) to IPng datagrams. While the option is algorithm-independent and will support many different authentication techniques, the use of keyed MD5 is proposed to help ensure interoperability within the worldwide Internet. This can be used to eliminate a significant class of network attacks, including host masquerading attacks. The use of the IPng Authentication Header is particularly important when source routing is used with IPng because of the known risks in IP source routing. Its placement at the internet layer can help provide host origin authentication to those upper layer protocols and services that currently lack meaningful protections. This mechanism should be exportable by vendors in the United States and other countries with similar export restrictions because it only provides authentication and integrity, and specifically does not provide confidentiality. The exportability of the IPng Authentication Header encourages its widespread deployment and use.

The second security option provided with IPng is the "IPng Encapsulating Security Header." This mechanism provides integrity and confidentiality to IPng datagrams. It is simpler than some similar security protocols (e.g., SP3D, ISO NLSP) but remains flexible and algorithm-independent. To achieve interoperability within the global Internet, the use of DES CBC is being used as the standard algorithm for use with the IPng Encapsulating Security Header.

Transition mechanisms

The key transition objective is to allow IPng and IPv4 hosts to interoperate. A second objective is to allow IPng hosts and routers to be deployed in the Internet in a highly diffuse and incremental fashion, with few interdependencies. A third objective is that the transition should be as easy as possible for end-users, system administrators, and network operators to understand and carry out.

The *Simple IP version Six Transition* (SIT) is a set of protocol mechanisms implemented in hosts and routers, along with some operational guidelines for addressing and deployment, designed to make transitioning the Internet to IPng work with as little disruption as possible [20].

SIT provides a number of features, including:

- *Incremental upgrade and deployment:* Individual IPv4 hosts and routers may be upgraded to IPng one at a time without requiring any other hosts or routers to be upgraded at the same time. New IPng hosts and routers can be installed one by one.

- *Minimal upgrade dependencies:* The only prerequisite to upgrading hosts to IPng is that the DNS server must first be upgraded to handle IPng address records. There are no prerequisites to upgrading routers.

- *Easy Addressing:* When existing installed IPv4 hosts or routers are upgraded to IPng, they may continue to use their existing address. They do not need to be assigned new addresses. Administrators do not need to draft new addressing plans.

- Low start-up costs. Little or no preparation work is needed in order to upgrade existing IPv4 systems to IPng, or to deploy new IPng systems.

continued on next page

IP Next Generation Overview *(continued)*

The mechanisms employed by SIT include:

- An IPng addressing structure that embeds IPv4 addresses within IPng addresses, and encodes other information used by the transition mechanisms.
- A model of deployment where all hosts and routers upgraded to IPng in the early transition phase are "dual" capable (i.e., implement complete IPv4 and IPng protocol stacks).
- The technique of encapsulating IPng packets within IPv4 headers to carry them over segments of the end-to-end path where the routers have not yet been upgraded to IPng.
- The header translation technique to allow the eventual introduction of routing topologies that route only IPng traffic, and the deployment of hosts that support only IPng. Use of this technique is optional, and would be used in the later phase of transition if it is used at all.

SIT ensures that IPng hosts can interoperate with IPv4 hosts anywhere in the Internet up until the time when IPv4 addresses run out, and allows IPng and IPv4 hosts within a limited scope to interoperate indefinitely after that. This feature protects the huge investment users have made in IPv4. SIT ensures that IPng does not render IPv4 obsolete. Hosts that need only a limited connectivity range (e.g., printers) need never be upgraded to IPng.

The incremental upgrade features of SIT allow the host and router vendors to integrate IPng into their product lines at their own pace, and allows the end users and network operators to deploy IPng on their own schedules.

Why IPng?

There are a number of reasons why IPng is appropriate for the next generation of the Internet Protocol. It solves the Internet scaling problem, provides a flexible transition mechanism for the current Internet, and was designed to meet the needs of new markets such as nomadic personal computing devices, networked entertainment, and device control. It does this in a evolutionary way which reduces the risk of architectural problems.

Ease of transition is a key point in the design of IPng. It is not something was was added in at the end. IPng is designed to interoperate with IPv4. Specific mechanisms (embedded IPv4 addresses, pseudo-checksum rules etc.) were built into IPng to support transition and compatibility with IPv4. It was designed to permit a gradual and piecemeal deployment with a minimum of dependencies.

IPng supports large hierarchical addresses which will allow the Internet to continue to grow and provide new routing capabilities not built into IPv4. It has cluster addresses which can be used for policy route selection and has scoped multicast addresses which provide improved scalability over IPv4 multicast. It also has local use addresses which provide the ability for "plug and play" installation.

The address structure of IPng was also designed to support carrying the addresses of other internet protocol suites. Space was allocated in the addressing plan for IPX and NSAP addresses. This was done to facilitate migration of these internet protocols to IPng.

IPng is designed to have performance better than IPv4 and still work well in low bandwidth applications like wireless. Its headers are less expensive to process than IPv4 and its 128-bit addresses are chosen to be well matched to the new generation of 64-bit processors. Its compact header minimizes bandwidth overhead which makes it appropriate for wireless use.

IPng provides a platform for new Internet functionality. This includes support for real-time flows, provider selection, host mobility, end-to-end security, auto-configuration, and auto- reconfiguration.

In summary, IPng is a new version of IP. It can be installed as a normal software upgrade in internet devices. It is interoperable with the current IPv4. Its deployment strategy was designed to not have any "flag" days. IPng is designed to run well on high performance networks (e.g., ATM) and at the same time is still efficient for low bandwidth networks (e.g., wireless). In addition, it provides a platform for new internet functionality that will be required in the near future.

Additional information

A set of World-Wide Web pages is available describing IPng. The URL for these is:

`http://playground.sun.com/pub/ipng/html/ipng-main.html`

These Web pages contain pointers to current specifications and implementations and will be updated on a regular basis.

The documentation listed in the reference sections can be found in one of the IETF Internet Draft directories.

In addition other material relating to IPng (such as *PostScript* versions of presentations on IPng) can also be found in the IPng working group archive:

`ftp.parc.xerox.com` in the `/pub/ipng` directory.

To join the IPng working group, send an electronic mail message to:

`majordomo@sunroof.eng.sun.com`

with "`subscribe ipng`" in the body portion of the message.

An archive of mail sent to this mailing list can be found in the IETF directories at `cnri.reston.va.us`.

References

[1] R. Hinden, Editor, "IP Next Generation Addressing Architecture," Internet Draft, October 1994.

[2] Y. Rekhter, P. Lothberg, "An IPv6 Global Unicast Address Format," Internet Draft, January 1995.

[3] Y. Rekhter, T. Li, "An Architecture for IPv6 Unicast Address Allocation," Internet Draft, February 1995.

[4] R. Atkinson, "IPv6 Authentication Header," Internet Draft, August 1994.

[5] S. Thomson, "IPv6 Address Autoconfiguration," Internet Draft, February 1995.

[6] V. Fuller, et al, "Supernetting: an Address Assignment and Aggregation Strategy," RFC 1338.

[7] W. Simpson, "IPv6 Header Compression," Internet Draft, September 1994.

continued on next page

IP Next Generation Overview *(continued)*

[8] W. Simpson, "IPv6 Neighbor Discovery—Processing," Internet Draft, November 1994.

[9] W. Simpson, "IPv6 Neighbor Discovery—ICMP Message Formats," Internet Draft, January 1995.

[10] S. Thomson, "IPng Extensions to BOOTP/DHCP," Internet Draft, In Preparation.

[11] S. Thomson, C. Huitema, "DNS Extensions to support IP version 6," Internet Draft, October 1994.

[12] C. Huitema, "The H Ratio for Address Assignment Efficiency," Internet Draft, In Preparation.

[13] S. Deering, A. Conta, "ICMP and IGMP for the Internet Protocol Version 6 (IPv6)," Internet Draft, In Preparation, October 1994.

[14] R. Hinden, Editor, "Internet Protocol, Version 6 (IPv6) Specification," Internet Draft, October 1994.

[15] J. Postel, "Internet Protocol," RFC 791, September, 1981.

[16] S. Bradner, A. Mankin, RFC 1752, "The Recommendation for the IP Next Generation Protocol," January 1995.

[17] R. Atkinson, "IPv6 Security Architecture" Internet Draft, September 1994.

[18] R. Atkinson, "IPng Encapsulating Security Payload (ESP)," Internet Draft, In Perpetration.

[19] S. Deering, "Simple Internet Protocol Plus (SIPP) Specification (128-bit address version)," Internet Draft, July 1994.

[20] R. Gilligan, "Simple IPv6 Transition (SIT) Overview, Internet Draft, November 1994.

[21] J. Reynolds & J. Postel, "Assigned Numbers," RFC 1340, July 1992.

[22] D. Crocker, "The ROAD to a New IP," *ConneXions,* Volume 6, No. 11, November 1992.

[23] F. Solensky, "Continued Internet Growth," Proceedings of the Eighteenth Internet Engineering Task Force, August 1990.

[24] F. Solensky, "The Growing Internet," *ConneXions,* Volume 6, No. 5, May 1992.

[25] *ConneXions,* Special Issue: IP—The Next Generation,Volume 8, No. 5, May 1994.

More about IPng:

Sessions: IE5, IE6 & C23
Workshop: W804

See Program Guide for details.

ROBERT M. HINDEN holds an B.S.E.E., and a M.S. in Computer Science from Union College, Schenectady, New York. Mr. Hinden is Director of Software at Ipsilon Networks, Inc. of Mountain View, California. Ipsilon Networks is a startup working in the area of high performance internetworking and ATM. Mr. Hinden was previously employed at Sun Microsystems where he was responsible for the department which develops internet protocols for Sun's operating systems. Prior to this he worked at Bolt, Beranek, and Newman, Inc. on a variety of internetwork related projects including the first operational internet router and one of the first TCP/IP implementations. He has been active in the IETF since 1985 and is currently the document editor for the IPng working group and a member of the IPng Directorate. He served as the Area Director for Routing in the Internet Engineering Steering group from 1987 to 1994 and previously co-chaired the Simple Internet Protocol Plus working group and chaired the IP over ATM and the Open Routing working groups. He can be reached on the Internet at `hinden@ipsilon.com`

IPng—What it means for OSI

by Tim Dixon, TDC Network Consultancy

Introduction

In March 1992, the Routing and Addressing group of the *Internet Engineering Task Force* (IETF) reported its conclusions on the fundamental architectural problems of the network-layer protocol of the Internet (IP) which were being exposed by the rapid growth in the number of Internet-connected networks. The group made short-term recommendations to alleviate operational difficulties, but also pointed to the longer-term prospect of the complete exhaustion of the 32-bit IP address space. It therefore called for work to begin on exploring ways of expanding the IP address space to cope with the expected growth in the number of networked computer systems.

Shortly afterwards, the *Internet Architecture Board* (IAB) (the "generals" as opposed to the IETF "foot soldiers") issued their opinion that the shortage of address space could be solved by using the OSI *Connectionless Network Protocol* (CLNP) as the basis for an IP replacement. Those who had invested in DECnet/OSI® or other CLNP-based networking systems congratulated themselves on their foresight. The IETF, however, quickly realised that the IAB decision deprived them of a once-in-a-lifetime opportunity to design a fundamental new networking protocol and hence of an opportunity for technical immortality. The foot soldiers mutinied, the generals were deposed, and for over two years protocol experts arranged and re-arranged boxes labelled "0" to "31" on countless sheets of paper and overhead transparencies, seeking the perfect pattern.

Finally, in the latter part of 1994, a recommendation [1] emerged from all this effort for an *IP Next Generation Protocol* (IPng) which is sufficiently similar in technical respects to CLNP for the IAB to claim that they knew what they were doing all along, and yet sufficiently different in detail for the IETF to claim victory.

So, it seems that the the early adopters of CLNP congratulated themselves too soon. Where does that leave those with OSI-based networks who were hoping for a convergence of wide-area networking technologies? In order to answer that question, a brief look inside IPng is required.

Features of IPng

Amongst the requirements for IPng were that it should be able to scale to the interconnection of 1,000,000,000 separate networks, provide security and extensibility and be simple to configure: in short that it should be possible to build a universal network without requiring universal expertise. Whether all the goals are in fact met depends in some measure on development work that is still to take place, but the main features of IPng are now known.

The recommendation for IPng is based on work which originally went under the title *Simple Internet Protocol* and indeed the main feature of IPng is that it bears a striking similarity to the existing IP with some of the more baroque detail removed. Major differences are:

- 128-bit network-layer addresses
- Simplified header format
- Practical source-specified routes
- Authentication, Integrity and Confidentiality at the network layer
- Elimination of fragmentation and reassembly

continued on next page

IPng and OSI *(continued)*

One of the main arguments about IPng has been the size of the network-layer address. Although a process of rational argument can show that 64 bits are sufficient (and the same argument can show that 32 bits are not), delegation of assignment authority tends to "waste" address space in the interests of administrative simplicity. Coupled with the large installed base of Novell IPX-based networks (which have addresses of just a few more than 64 bits) the final recommendation of 128 bits is pragmatic. Interestingly, although the router manufacturers were not fazed by the prospect of variable-length addresses, developers of end-systems (nowadays principally RISC-based) argued forcefully for the fixed alignment of fields on 32-bit or 64-bit boundaries.

The simplified header format reduces the amount of processing that needs to be done at intermediate hops along the path. Changes in the way that source-routing is handled means that particular paths can be chosen by the source through various administrative boundaries without the need to identify every intermediate router along the way.

The provision of Authentication, Integrity and Confidentiality at the network layer answers many of the concerns that have been raised about the security of Internet-connected networks. Although it is intended that implementation of these features be mandatory, the effect of public policy, particularly in the area of confidentiality, may mean that they cannot be used in many areas of the world or that implementations will be subject to export controls which apply to cryptographic systems.

Fragmentation in the network layer has long been deprecated, since it tends to have a negative effect on both throughput and effective utilisation of network links. In IPng all datalinks must be capable of transmitting and receiving packets of at least 576 octets (using datalink-layer protocols to achieve this if the underlying physical medium cannot) and a process of "discovery" will be defined to permit the transport layer to adapt to the largest currently-available transmission unit.

Why should OSI-users care about IPng?

Use of OSI, particularly in Europe, is still in many circumstances mandated by public policy. In other cases, it is a consequence of the purchase of a particular vendor-solution (such as DECnet), or the availability of public services (such as X.400). This article concerns itself with the connectionless variant of the OSI network-layer protocol (CLNP), used by an estimated 500 to 800 thousand computer systems throughout the world [2]. Although similar economic considerations apply to the connection-oriented network service (typically provided over X.25), different technical aspects rule it out of consideration in this brief discussion.

The rapid growth of IP both as an internal networking system within businesses and as a means of access to public Internet services has several consequences for OSI users:

- The predominance of IP will lead to higher purchase and support costs for lower-volume OSI-based products and services.
- The available choice of equipment and applications may become increasingly constrained and trained staff increasingly scarce.
- Business needs for specific services may mandate access to the Internet, whatever the "in-house" networking protocol.

The true test of an "Open" network is universal connectivity, and clearly OSI is by now a long way behind IP in terms of interoperating systems installed. The removal of the 32-bit addressing limit on IP means that OSI users should consider the "Betamax scenario" and reflect on whether their long term needs will be met by OSI. There is clearly no immediate urgency to ditch existing investment: a network which is meeting a business need today will still do so tomorrow. For some organisations, a more pressing need may be the harmonisation of PC-LAN protocols with their wide area networks. Even those with an immediate requirement for some form of Internet connectivity may need no more than an electronic mail gateway right now. However, a review of OSI policy should figure in at least the medium-term plans of network managers. As networked applications and new business needs evolve, a likely future scenario is a mixed-protocol network supporting OSI, Internet and PC-LAN applications with an increasing emphasis on IPng as OSI maintenance becomes less cost-effective and IP systems are gradually upgraded.

Evolution of the mixed-protocol network

Current mixed-protocol networks may be based on a single routing scheme which supports multiple address- and packet-formats (e.g., variants of the ISO IS–IS intra-domain routing protocol which also support IP) or on independent routing schemes for each protocol, perhaps sharing the same physical topology by multiplexing or encapsulation, but which are essentially separately-administered networks.

If you have a network of integrated multiprotocol routers, you may wish to question your vendor closely about his plans to add IPng, as this may be the quickest and easiest means of introducing IPng support. Otherwise, careful thought should be given to the costs and consequences of running yet another routing protocol (even if it is only during a period of transition) in the network backbone. Since IP will remain a viable protocol long after the introduction of IPng (the aim being only to achieve transition before the IP address space runs out—currently projected to be some time after 2005 [1]), this is not a prospect that must be faced immediately.

However, network managers are more likely to be attracted by the prospect of a reduction in the number of backbone protocols rather than an increase, for example by using IPng as a carrier in which other networking protocols can be transported. Unfortunately, although IPng can reasonably be expected to carry IP and IPX, because OSI NSAPs can be up to 20 octets long and IPng addresses only 16, CLNP into IPng won't go. The blunt truth is that, although the Novell installed base is a powerful argument for going from 64 to 128 bits, the OSI installed base is not a powerful argument for going any further (although conspiracy theorists might suspect an IETF plot...). Two activities are underway to remedy this difficulty: to define a standard for "squeezing" NSAPs in commonly-used formats into IPng addresses and to define a way of representing IPng addresses as NSAPs [2, 3, 4]. There is still a final twist, though: the removal of fragmentation and reassembly in IPng means that even if the addresses can be handled, the transport of CLNP in IPng must not result in IPng packets larger than 576 bytes. It is unlikely that carrying CLNP within IPng can be regarded as anything other than an interim measure in a plan to phase out CLNP altogether.

Evolution of endsystems and applications

Whatever the problems of supporting IPng within the network backbone, far more difficulties result from attempting to integrate applications and core network services.

continued on next page

IPng and OSI *(continued)*

A network containing DECnet, Novell and IP will also contain a Digital Naming Service, a Novell Directory Service and an Internet Domain Name System. It may provide remote access to files using the DEC Distributed File System, the IP-based Network File System, Internet FTP and OSI FTAM. Some endsystems on the network may support only DECnet or Novell or Internet protocols, others may support all of them. Systems and services will be present in a variety of different databases, with different names, addresses and capabilities. Some models of the transition process from IP to IPng additionally require endsystems to support both IP and IPng protocols. Network managers are right to fear an administrative nightmare: some are no doubt already experiencing one. Just as it is now recognised to be a "bad thing" to allow users an unrestricted choice of hardware and software in organisations with large numbers of PCs, it is becoming obvious that much firmer policies may be needed in corporate networking provision. The advent of IPng is a good excuse to start developing firm policies if they don't already exist.

Commercial users will have to look to vendors, service-suppliers and their own resources to help bring some order to this chaos and adjust their budgets accordingly. Non-commercial users may just have to plan on a little less sleep for a while. If you have a particularly complex mixed network, it may well pay to take a long look at the lifetime and cost/benefit of current applications to see if some can be eliminated or migrated to require a smaller number of network protocols during the remaining lifetime of IP. Considering a network consisting only of two protocols, OSI and IP, the options are:

- Plan on a mixed OSI/IPng network for the future
- Plan on an IPng-only network, moving both OSI and IP to IPng in the future
- Move from OSI to IP now and from IP to IPng in the future

Implementations are already available of the OSI applications in widespread use (X.400 being the most common, with some vestiges of X.500 and FTAM) which support IP (using the procedures of RFC 1006) as well as the connectionless and connection-oriented variants of the OSI network service. Where these are in use, it may be possible to diminish the scope of network-layer OSI protocols within a mixed-protocol network by suitable re-configuration of the applications. If this is feasible, or if equivalent IP-based applications can be obtained, it may be possible to consider moving away from OSI now. Remember, though, that an IP to IPng transition will also be a major effort, so all pain is not eliminated by this process: the main reason for considering an OSI to IP move now would be if there were the possibility of immediate cost savings (for example from the elimination of duplicate naming services or administrative and support staff).

However, there will be many cases in which a significant investment in OSI-only equipment or applications mandates continued OSI network-layer connectivity although newer purchases require IP. In this case, the network manager has to decide how long to retain OSI, and specifically whether it will be retained even after there is a need to move to IPng. At the very least, network managers should be collecting protocol-usage statistics so that trends in usage can be monitored.

Watch this space

IPng is still very much "work in progress." Features such as auto-configuration (which provides "plug and play" capabilities) and support for mobile computer systems are still under development.

Transition plans from IP and IPng are also being written [5], and there is active interest in OSI to IPng migration. More details will become clear as specifications are written and reviewed, and anyone who has a particular interest can participate in the IETF process. The electronic mailing list concerned with OSI and IPng can be joined by sending an e-mail message with the string "`SUBSCRIBE NOSI`" to: `majordomo@sunroof.eng.sun.com`

There is periodically talk of ISO working on a new version of CLNP, and of convergence with IPng in the future. Although the technical need for such work may be moot, it is proving rather hard to shift public procurement policy in Europe away from OSI, despite its increasing relative deployment cost. There is now liaison between the IETF and relevant international standards bodies, and a possible mechanism by which IETF standards might be recognised through formal processes and thereby get round the political roadblock. In the meantime, anyone subject to public procurement rules should probably formulate their plans with judicious attention at least to the letter of the law.

Some the following documents are Internet Drafts and have a short lifetime. They can be retrieved via anonymous Internet FTP from several repositories.

References

[1] Bradner, S. & Mankin, A., RFC 1752, "The Recommendation for the IP Next Generation Protocol," January 1995.

[2] Carpenter, B., "Discussion Paper for NOSI BoF," Internet Draft, 1994.

[3] Carpenter, B., "Recommendations for OSI NSAP usage in IP6," Internet Draft, 1994.

[4] Houldsworth, J., "OSI NSAP usage in IP6," Internet Draft, 1994.

[5] Gilligan, R., "Simple IPv6 Transition (SIT) Overview, Internet Draft, November 1994.

[6] Thomson, S. & Huitema, C., "DNS Extensions to support IP version 6," Internet Draft, October 1994.

[7] R. Hinden, Editor, "Internet Protocol, Version 6 (IPv6) Specification," Internet Draft, October 1994.

[8] Crocker, D., "The ROAD to a New IP," *ConneXions,* Volume 6, No. 11, November 1992.

[10] Solensky, F., "The Growing Internet," *ConneXions,* Volume 6, No. 5, May 1992.

[11] *ConneXions,* Special Issue: IP—The Next Generation,Volume 8, No. 5, May 1994.

[Ed. This article is reprinted with permission from *Open Systems Networking and Computing* (OSN), Volume 9, Issue 1, January 1995. OSN is published by Technology Appraisals Ltd., London. For more information, send e-mail to: `techapp@cix.compulink.co.uk`].

More about IPng:

Sessions: IE5, IE6 & C23
Workshop: W804

See Program Guide for details.

For the last three years, **TIM DIXON** has been a consultant with TDC Network Consultancy based in Newcastle-upon-Tyne, UK and its affiliate, WebWork. Until the end of 1994, he was also secretary of the Technical Committee of RARE, the Amsterdam-based association of European Research Networks. Previously, he worked for Digital Equipment Corporation for over eight years developing X.25 software and providing the first implementation of the DECnet/OSI routing protocol. Tim has observed the development of IPng from the outset and is author of a number of papers on the subject, including RFC 1454. He has a BA from the University of Cambridge. E-mail: `dixon@webwork.co.uk`

Providing Interoperable High Speed Connections with Fibre Channel Technology

by
Peter Walford, DemoGraFX, Inc.
and
Ed Frymoyer, Hewlett-Packard

Introduction

Fibre Channel [1] is the new interconnect technology that constitutes a fusion of networking and channel approaches to device and system interconnection. This fusion allows system designers to combine traditional peripheral connection, host-to-host internetworking, loosely-coupled processor clustering and emerging multi-media applications in a single multi-protocol interface. The types of channel-oriented facilities incorporated into the Fibre Channel framing protocol include data-type qualifiers for routing frame payload into particular interface buffers and link-level constructs associated with individual I/O operations. The types of networking-oriented facilities include full multiplexing of traffic between multiple destinations, peer-to-peer connectivity between any pair of ports on a Fibre Channel Fabric, and capabilities for internetworking to other connection technologies. Depending on the needs of the application, either channel or networking approaches can be used for any data transfer. Supported classes of service include dedicated connections (circuit switching—Class 1) and frame-level packet switching (Classes 2 and 3); isochronous capabilities including constant latency or guaranteed availability fractional bandwidth are part of a new Class of Service (Class 4) being incorporated into the extension of the Fibre Channel Standard, FC-EP. Nodes connect to Fibre Channel via adapter-level interfaces known as *N_Ports*.

Products employing Fibre Channel are now being installed at the rate of hundreds of new connection paths each day in commercial, engineering, scientific, and graphical applications. Both mass storage connections as well as clustered connections are being deployed, with applications beginning to see the commercial benefit of Fibre Channel linkages at rates of 266 to 1062 Mbps with latencies in 10's of microseconds.

Profiles

In order to foster applications built on essential common features, lower costs, develop multiple suppliers and most importantly to ensure interoperability among suppliers, the first *Fibre Channel Profiles* were developed by the *Fibre Channel Systems Initiative* (FCSI). FCSI, formed in February 1993 by Hewlett-Packard, Sun Microsystems and IBM (RISC 6000 Division), is developing Profiles to serve as high level design specifications for the development of interoperable Fibre Channel systems. The broad industry based *Fibre Channel Association* (FCA), which was formed in late 1993 including the FCSI member companies and now has over 60 members worldwide, is also developing Profiles for Directory Services, IPI and Fabric interoperability issues. An industry ad-hoc group has developed the Fibre Channel Arbitrated Loop Direct Attach Disk Profile [10] as an interoperability specification for direct attachment of individual disk drives to hosts and RAID systems using Fibre Channel. The goal of these profile efforts is to ensure a rapid start to the Fibre Channel industry by providing interoperability from the first or second generation of equipment.

FCSI assists in the development of Interoperability Testing procedures and provides technical assistance and liaison for industry interoperability testing centers. FCSI also has a major role in education about the benefits, attributes, and use of Fibre Channel technology and has staged interoperability demonstrations at 1994 Interops and will do so again in 1995 as ever more products come on line.

Topologies

Fibre Channel configurations cover three major types of topologies:

- *Point-to-point attachment:* Devices are connected directly to each other over a dedicated medium.
- *Switched Fabric:* Devices are connected to ports on a switching Fabric. This switched topology provides scalability of bandwidth: as additional ports are added, the aggregate bandwidth of the Fabric increases, thereby minimizing congestion and contention and increasing throughput.
- *Arbitrated Loop:* A low-cost shared-media topology, providing gigabit connections for small configurations and device attachment without the cost of a Fibre Channel Switch. Arbitrated Loops can be integrated into switched Fabrics, providing connectivity between hosts and devices on loops and those directly connected to switching Fabric ports.

The type of topology is discovered early in the data link initialization process.

The Fibre Channel standard defines a broad range of media types, including shortwave laser, longwave laser, copper and LED options at various speeds (Fibre Channel is the only draft networking standard with a fully-specified gigabit data link-level media option) and various distances. All share a common 8B/10B encoding scheme for data and signal integrity in different environments. The standard includes facilities ranging from the media connect through signaling and framing protocol specifications culminating in a Service Interface for use by Upper Level Protocols. The Fibre Channel standard family includes mapping standards (called FC-4s) for mapping Upper Level Protocol operations and services to the Fibre Channel communications architecture. For example, there are mapping layers for SCSI [2] and for IP [4].

Nowadays, the success of computer technologies in the marketplace is determined by the technology's ability to enable new applications or by the increase in functionality or performance that the technology brings to existing applications. This increased functionality or performance (ex. rate) must be provided at the same cost as existing technologies and with minimal changes to existing interfaces. Accordingly, the FCSI Profiles are not limited to transport-level interoperability issues. Instead, the Profiles include option selections from the protocol-mapping layers, application-specific command sets, and any other intermediate standards required to ensure application-level interoperability. There are specific Profiles for individual applications, including a SCSI Profile for interoperability between Fibre Channel-attached SCSI devices, an IP Profile for IP host-to-host internetworking, etc.

In early implementations of a new technology, interoperability problems can result from minor differences in implementation of complex facilities. Complexity increases the cost and hinders the development of commodity components necessary for the growth of a volume market.

continued on next page

Fibre Channel Technology *(continued)*

A major goal in the Profile development process was agreeing upon common simple mechanisms that could be used in multiple applications, on different topologies and in different Classes of service which simultaneously enabled a broad third party supply base.

Options

The FCSI Profiles achieve this goal of simplicity through a process of option reduction. The FCSI technical teams examined each option and each type of behavior allowed in the relevant standards to determine the answers to several questions:

- What is the effect of the option on basic interoperability (i.e., if one N_Port supports the option and the communicating N_Port doesn't, can the two N_Ports interoperate)?
- Does the option mandate a particular type of implementation?
- What are the potential mechanisms that can be used to perform the function?
- What is the level of complexity required in one N_Port to interoperate with another N_Port that implements the function differently?

Simplicity of implementation safeguards against interoperability problems resulting from minor errors in implementation by different vendors. It also facilitates testing: to verify interoperability, all possible combinations should be tested, and complex scenarios involving many different ways of implementing the same function require a lot of testing.

The three FCSI partners had very different perspectives and approaches to Fibre Channel implementations. Consequently, the collective expertise of the three companies represented a broad spectrum of Fibre Channel technology and methodology. This ensured that FCSI's consideration and evaluation of various techniques was thorough and comprehensive, as each company would often favor a different approach. Whenever possible, FCSI would select only one of several alternative techniques, as interoperability with multiple techniques might require all implementations to include each method, at a greater risk to interoperability and at an increase in end-user cost.

The Profiles constitute a mutually agreed-upon proper subset of the various standards involved in a particular application. However, there is considerable latitude for a wide range of cost/performance tradeoffs: the Profiles provide for interoperability between low-cost medium-performance implementations and more expensive implementations with performance-enhancing features.

FCSI has currently published the following Profile documents:

- *Profile Structure* [5]. Describes the lexical elements and organization of FCSI Profiles.
- *Common Feature Sets* [8]. Describes features common to multiple FCSI Profiles, including most signaling protocol features.
- *SCSI Profile* [6]. Interoperability specification for SCSI hosts and Fabric-attached multi-drive mass storage subsystems.
- IP Profile [7]. Interoperability specification for IP hosts.
- *Gigabaud Link Module Specification* [9]. Component-level specification for a family of physical interfaces.

Examples

The types of criteria used in selecting features for the Profiles and the interoperability issues involved are best explored through examples. This section includes several technical examples illustrating features designed to eliminate basic interoperability problems, selection of features from different architectural layers required to ensure end-to-end application interoperability, and the Profiles' role in forestalling problems commonly found in early implementations of new technologies by defining a simple and comprehensive error recovery mechanism.

Fibre Channel constructs

Selected facilities from the Fibre Channel architecture are described here to facilitate understanding of the following examples. Specifically, the organization of Fibre Channel frames into Sequences and Exchanges is discussed, along with types of acknowledgment frames and flow control.

Frames, Sequences and Exchanges

Although the basic unit of data transferred on a Fibre Channel link is a *Frame* carrying between 128 and 2112 bytes of payload, the information unit presented at the interface to the signaling (framing protocol) level is a *Sequence* consisting of a variable number of frames. A Sequence is a contiguous block of data meaningful to the particular Upper Level Protocol; the Fibre Channel signaling level segments a Sequence into frames for transmission and performs Sequence reassembly upon reception.

Sequences are themselves grouped into *Exchanges,* which provide a mechanism for organizing multiple Sequences into higher-level constructs for the convenience of applications. For example, in the *Fibre Channel Protocol for SCSI* (FCP [2]) the various command, data transfer and response Sequences associated with an individual disk operation are grouped into a single Exchange, allowing the higher-level SCSI software to treat all the transport functions of the operation as a single atomic unit for tracking and error recovery purposes. Similarly, an RPC-based client–server application could treat an individual RPC call as a single Exchange, with the request, parameter transfer and return value transport functions as separate Sequences.

Acknowledgment primitives

At the signaling level, Fibre Channel includes three different types of acknowledgment frame:

- *ACK_0:* a single frame used to acknowledge an entire Sequence
- *ACK_1:* a frame that acknowledges an individual data frame
- *ACK_N:* a frame that acknowledges one or more data frames

These acknowledgment types cannot be mixed. For example, if ACK_1 is used, then every data frame must be acknowledged by an individual ACK_1 frame; and if ACK_0 is used, then one and only one acknowledgment frame shall be transferred for an entire Sequence upon successful reception of the entire Sequence. During initialization prior to exchanging data, the Fibre Channel architecture provides for a login discovery process in which a pair of N_Ports may determine which facilities are mutually supported. During this login process, the type of acknowledgment that will be used in each Class of Service for all communications between two N_Ports is determined.

continued on next page

Fibre Channel Technology *(continued)*

Flow Control

Fibre Channel provides two flow control mechanisms to prevent data over-run and assist in congestion management:

- *End-to-End Flow Control:* This type of flow control is exercised between the communicating N_Ports. Depending on its buffering capabilities, the receiving N_Port provides credit for a certain number of frames to the transmitter; the transmitting N_Port limits the number of unacknowledged outstanding frames to this allocated credit and regulates its transmission rate according to the rate of acknowledgments received. This is the only type of flow control available on dedicated Connections.
- *Buffer-to-Buffer Flow Control:* This type of flow control regulates the traffic of individual frames between interface buffers at each stage of a frame's progress through the packet-switching facilities of a Fabric. Special signaling primitives indicate the availability of a buffer in the next stage of the path, and a frame is held in the current buffer until the next buffer is available. This prevents data overrun at destination N_Ports and at each switching element along the path; however, it is less effective at reducing congestion than the End-to-End mechanism.

Acknowledgment types

The interaction between Fibre Channel's flow control mechanisms, classes of service and acknowledgment types required careful study within FCSI. For example, the ACK_0 acknowledgment type could not provide interoperability between N_Ports with different levels of on-board buffering in Class 1. Figures 1 and 2 show the relationship between on-board buffering and flow control when using dedicated Connections.

Frames received into adapter memory must be eventually transferred across the system bus into host memory. If on-board memory is available, an adapter should be able to receive frames at link rate from the Fibre Channel. Flow control is not necessary as long as there is sufficient on-board memory to receive the incoming data. However, the adapter may be unable to transfer frames at link rate from on-board buffers to host memory due to system-level bus contention, context switching overhead, and other activity in the host. The transmission of acknowledgment frames allows a recipient to notify the transmitter that on-board buffers are available to receive incoming frames; the recipient can withhold those acknowledgments when on-board buffers are full pending transfer of received frames to host memory.

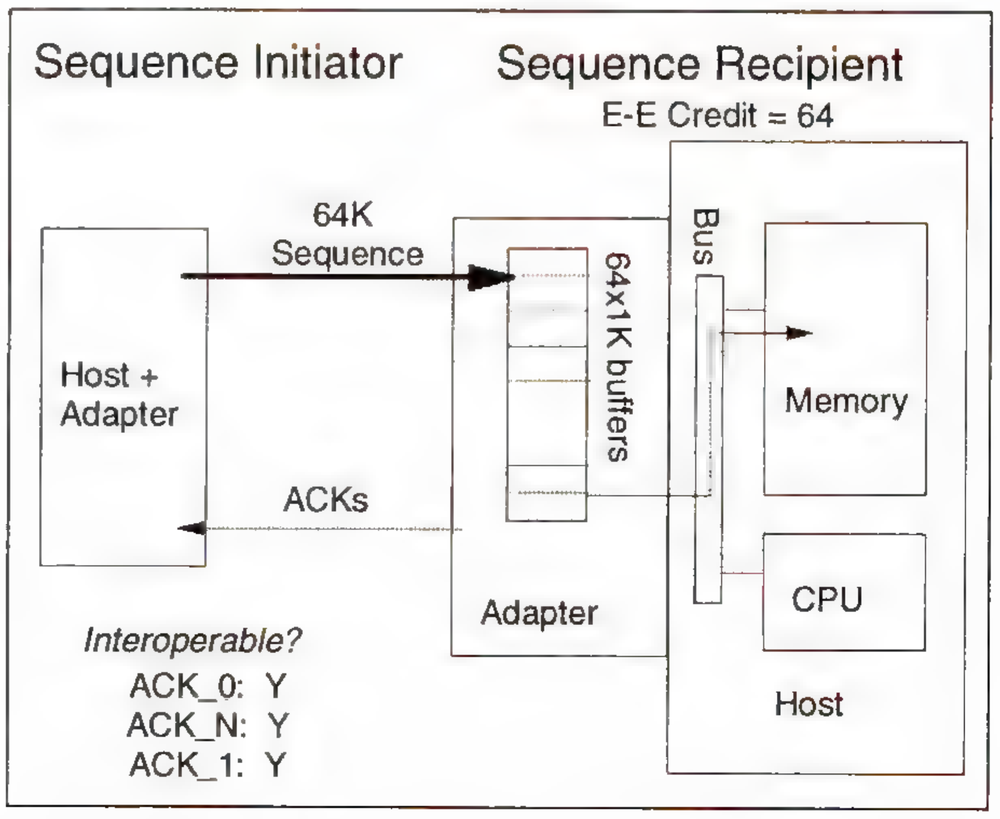


Figure 1: Flow control with large on-board buffering

Figure 1 shows an adapter with sufficient on-board buffering to receive an entire 64 KByte Sequence transferred as 64 1K frames. As the entire Sequence can be received in the interface board prior to transferring it to host memory, End-to-End flow control is unnecessary and the entire Sequence can be acknowledged with a single ACK_0 frame. Even if each data frame is acknowledged with an individual ACK_1, the entire Sequence will still be transmitted without flow control as the End-to-End credit value is 64 buffers.

On the other hand, Figure 2 shows an adapter with only 2 1KByte buffers on-board. Once this adapter has received two frames, over-run can only be prevented if the transmitter refrains from sending additional frames until it receives an indication that the on-board buffers are again available. Consequently, an ACK_0 cannot be used to acknowledge the entire Sequence as ACK_0s provide no information for intra-Sequence flow control. If ACK_1 is used, then the transmitter will not send more than two frames without acknowledgment as the End-to-end Credit value is only two. As each frame is transferred to host memory, the receiving adapter sends an ACK_1 frame indicating than an on-board buffer is available, and the entire Sequence can be transmitted without over-run.

The FCSI Profiles should provide for interoperability across a range of price/performance optimizations in adapter design; and the level of on-board buffering is a major differentiator between adapters with different price/performance levels. Accordingly, use of ACK_0 is prohibited in Class 1 in Profile-compliant operations [8].

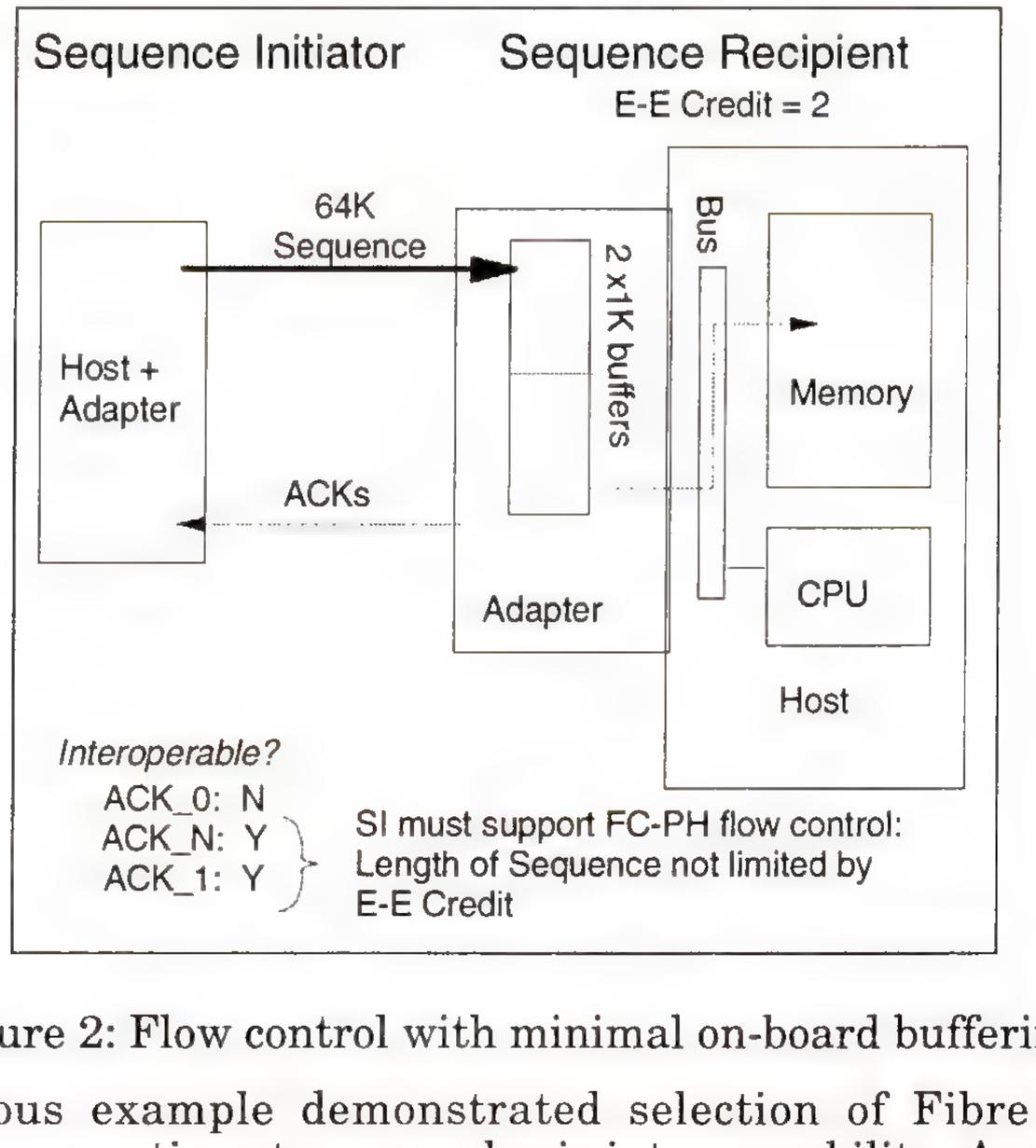


Figure 2: Flow control with minimal on-board buffering

TCP/IP on Fibre Channel

The previous example demonstrated selection of Fibre Channel signaling layer options to ensure basic interoperability. A major part of each Profile (and all of the Common Feature Sets document) is devoted to interoperability at the Fibre Channel level. However, as indicated previously, ensuring interoperable application support also involves option selection in Upper Level Protocols and among application features that use those protocols. The following example illustrates how the TCP/IP Profile [7] specification ensures that IP hosts implement compatible TCP/IP and ARP features to enable maximum throughput and connectivity.

continued on next page

Fibre Channel Technology *(continued)*

The implementation of TCP/IP on Fibre Channel is a straightforward process similar to its implementation on other LAN technologies. Mapping TCP/IP onto Fibre Channel constructs is defined in an IETF Internet Draft [4], which should be upgraded to proposed standard status in the near future.

The FCSI TCP/IP Profile stipulates that IP packets can be sent across Fibre Channel Fabrics in either the Connection-oriented Class of Service (Class 1) or the acknowledged Connectionless Class of Service (Class 2). In this context, the Connection-oriented Class of Service (Class 1) should not be confused with transport-layer TCP connections. A Fibre Channel Connection provides a dedicated path through a Fibre Channel Switching Fabric; all resources and link facilities along the path between two N_Ports are locked down at Connection establishment, providing guaranteed in-order delivery at full link bandwidth. Establishment and tear-down times for these Connections are very short (10–40 microseconds on current Fabrics), and for TCP/IP traffic these Connections will be very short-lived. In heavily utilized environments, a separate Fibre Channel Connection may be established for each IP packet.

The acknowledged Connectionless Class of Service (Class 2) provides for multiplexed traffic at the frame level (an IP packet may be sent as many 2K byte frames) with notification of delivery between multiple concurrently active N_Ports. Buffer-to-buffer flow control ensures that frames are not accepted into the Fabric unless switching buffers are available to route them; so that Fabrics can respond to temporary congestion by not accepting frames rather than discarding data due to statistically overcommitted bandwidth allocations.

Individual IP packets are sent as separate Fibre Channel Sequences. Many IP packets can be sent in a single Fibre Channel frame (maximum size if 2112 bytes); these are transmitted as single-frame Sequences and require no fragmentation or reassembly. Larger packets are sent as multi-frame Sequences, fragmentation into multiple frames and reassembly at the destination are performed by Fibre Channel protocol chips in the adapters. A large IP *Maximum Transmission Unit* (MTU) size (62580 bytes on Fibre Channel [4]) can be used as both Class 1 and Class 2 provide guaranteed delivery (on networks that respond to congestion by dropping frames or cells, large MTU sizes are not practical as the entire packet will need to be retransmitted if a small cell is dropped). In data-intensive applications, throughput is improved with a large MTU size: a host interrupt is required per packet, and fewer interrupts are required with larger packets.

Measured TCP/IP throughput on Fibre Channel is very impressive. For example, IBM has reported inter-workstation TCP throughput of 17 Mbytes/sec across a Fabric using standard 25 Mbytes/second Fibre Channel optical links, and there are no barriers to this throughput scaling up as gigabit interfaces become available later this year.

Performance recommendations

As Fibre Channel throughput is high, the FCSI TCP/IP Profile recommends that TCP implementations incorporate the TCP Options and Extensions for performance on high-speed LANs described in "TCP Extensions for High Performance" [12].

These extensions include the following mechanisms supported by TCP options negotiated at TCP connection establishment:

- *Window Scale:* TCP option that provides for increasing the TCP window of unacknowledged data from the standard limit of 64K bytes to up to 1G byte.
- *Accurate Round-trip Time measurement:* this increases the accuracy of round-trip time (the time interval between sending a TCP segment and receiving the acknowledgment) measurement at negligible computational cost on TCP connections using large windows on very high bandwidth networks. This mechanism requires the TCP Timestamp option.
- *Protection against Wrapped Sequences:* improves data integrity by protecting against inadvertent data corruption associated with delayed delivery of retransmitted data after a TCP Sequence Count wraparound has occurred (the probability of such wraparounds is much greater with high bandwidth networks). This mechanism also requires the TCP Timestamp option.

Address resolution

As with all LANs, address resolution capabilities are necessary to provide plug-and-play level TCP/IP interoperability on Fibre Channel. Address Resolution on Fibre Channel is provided by an adaptation of the standard ARP protocol; the general mechanism is described in the Internet Draft [4]. In combination with the FCSI TCP/IP Profile, ARP support is specified for the three Fibre Channel topologies as follows:

- *Point-to-Point:* each workstation responds to ARP requests sent from the directly attached node.
- *Fabric-attached:* Address resolution is provided by an ARP Server accessed via a well-known address on the Fabric. Each IP node registers its IP and Fibre Channel addresses with the ARP Server, and nodes needing IP-to-Fibre Channel address translation query the ARP Server for this information (this mechanism is very similar to that used by "Classical IP over ATM" [11]). The FCSI Profile specifies that these requests will always be performed in Class 2, simplifying ARP Server implementation.
- *Arbitrated Loop (FC-AL):* as this uses a shared medium similar to traditional LANs, broadcast support on this topology is easily implemented. ARP requests are sent out as broadcast frames, and the node with the requested IP address unicasts a response to the ARP request originator.

Error recovery

Robust, reliable error recovery is a critical component of interoperability. Frame corruption due to transmission errors occurs with any data communications technology, although the frequency of errors, error detection capabilities, and error correction and recovery facilities vary a great deal. The robustness of error detection and recovery facilities is often poorly understood during initial standard development; the experience and analysis gained from early implementation efforts is required to understand the most practical level of error recovery mechanisms to implement and evaluate the tradeoffs between frequency of errors, efficiency of recovery mechanisms, and cost and complexity of implementation.

Error recovery constituted a major focus of FCSI's Profile development efforts. It was particularly important to correctly determine the appropriate level of recovery and the degree of complexity and sophistication required.

continued on next page

Fibre Channel Technology *(continued)*

Standard conformant mechanisms ranged from resetting the physical link to immediate retransmission of Sequences containing errors to forced logout of an N_Port after communications failures. Successful error recovery with any of these mechanisms requires an exact interoperable match between the operations performed by one N_Port and those of the N_Port with which it is communicating.

To ensure interoperability among early implementations, a single simple error recovery mechanism sufficiently flexible for recovery from all anticipated errors was considered preferable to requiring a battery of more sophisticated mechanisms that might provide more efficient recovery from particular types of errors. More sophisticated mechanisms could be added in the future, after experience with early implementations and once basic heterogeneous interoperability had been achieved.

Errors in Fibre Channel environments are expected to be quite infrequent. This is due to the low bit error rate of the transmission media employed, the resiliency of the encoding scheme, and the minimization of congestion through the flow control mechanisms in the various Classes of Service. In this environment, simplicity of implementation and broad applicability of a recovery mechanism is more important than minimizing overhead. Maintenance of data integrity and reliable return to a predictable known state after errors are critical. The same mechanism should be available for Upper Level Protocol initiated error recovery (e.g., termination of a disk operation after an operation-specific time-out) as for data link level recovery.

The Fibre Channel architecture provides for error recovery at any of four levels:

- *Link level,* where errors in transmission of individual frames result in link-level resets.
- *Sequence level,* where errors (including frame-level errors) are corrected by abnormal termination and retransmission of the Sequence, if possible.
- *Exchange or operation level,* where upon any error the entire operation (possible involving many Sequences, each of which is composed of many frames) is aborted and retried according to the recovery requirements of the particular application.
- *N_Port level,* where recovery to a known state requires a temporary loss of the ability to communicate between two N_Ports followed by re-login.

These levels constitute a hierarchy of progressively more comprehensive recovery mechanisms. Link level recovery is necessary if link integrity is compromised; however, link resets can cause frames in transit to be discarded leading to a requirement for Sequence level recovery. Exchange level recovery can always correct Sequence level errors, but Sequence level recovery is insufficient if entire Sequences are lost or if an entire operation must be aborted at the request of higher level software. The N_Port level is the most drastic, and is only appropriate if other mechanisms have failed.

All N_Ports must implement link-level recovery to maintain integrity and synchronization along the transmission link. Of the four mechanisms, Sequence-level recovery is the most complex; as the state of individual Sequences must be tracked and adapters must be capable of retransmitting particular Sequences. Exchange level recovery would still be required even if Sequence-level recovery was available.

Profile-compliant error recovery occurs at the Exchange (i.e., operation) level; attempts to invoke Sequence-level recovery are prohibited. As errors should be infrequent and Exchange level recovery would always be required for certain types of errors, the selection of Exchange level recovery represents a significant simplification that ensures data integrity and robustness of Fibre Channel applications. For SCSI operations, Exchange-level recovery means that the entire I/O operation is aborted and may be retried. Error recovery in IP applications is always provided above the data link level (Fibre Channel for IP simply provides a transmission facility for datagrams of up to 64K bytes in length), Exchange-level error recovery simply restores the Fibre Channel transport mechanism to a known state. As most IP hosts might also use Fibre Channel for SCSI disk operations, the FCSI Profiles provide for an identical error recovery mechanism for IP and SCSI.

Error recovery specifications are neither complete nor interoperable unless errors that occur during the error recovery process are themselves addressed. The FCSI Profiles provide for limited attempts to repeat failed error recovery operations; followed by implicit logout of the destination N_Port if the retry fails. If communications with a particular N_Port have become impossible due to the frequency of errors encountered, this ensures that all operations with that N_Port are terminated. Communications can be re-established by re-login once the situation causing frequent errors has been corrected.

Conclusion

The FCSI Profiles represent a unique effort to ensure multi-vendor interoperability among early implementations of a complex new gigabit interconnect technology. These Profiles include selections among all options required for application-level interoperability. Through thorough specification of all interoperability features including error recovery, the Profiles provide a platform for the rapid development of Fibre Channel products to meet the growing bandwidth demands of data communications and mass storage in the workstation marketplace.

Acknowledgments

The FCSI Profiles that form the basis of this work are the combined product of the Fibre Channel technical teams of Hewlett-Packard, Sun Microsystems, Inc., and IBM (RISC/6000 System Division). The authors gratefully acknowledge the support of the Fibre Channel Systems Initiative in the preparation of this material.

References

[1] Fibre Channel—Physical and Signaling Interface (FC-PH), Revision 4.3, X3T11/Project 755D/Rev 4.3.

[2] Fibre Channel Protocol for SCSI (FCP), Revision 9, X3T10-993D.

[3] Small Computer Systems Interface—2 (SCSI-2), X3.131, 1994.

[4] "IP and ARP on Fibre Channel," Internet Draft, 1995.

[5] FCSI Profile Structure, FCSI-001—Revision 1.0.

[6] FCSI SCSI Profile, FCSI-201—Revision 2.1.

[7] FCSI IP Profile, FCSI-202—Revision 2.0.

continued on next page

Fibre Channel Technology *(continued)*

[8] FCSI Common FC-PH Feature Sets Used in Multiple Profiles, FCSI-101—Revision 3.0.

[9] Gigabaud Link Module Specification, FCSI-301 Revision 1.0.

[10] FC-AL Direct Attach Disk Profile (Private Loop), Version 1.3.

[11] Laubach, M., "Classical IP and ARP over ATM," RFC 1577, January 1994.

[12] Braden, R., & Jacobson, V., "TCP Extensions for High Performance," RFC 1323, May 1992.

[13] Plummer, D., "An Ethernet Address Resolution Protocol," RFC 826, November 1982.

[14] Postel, J., "Multi-LAN Address Resolution," RFC 925, October 1984.

[15] Atkinson, Ran, "Towards Real ATM Interoperability," *ConneXions,* Volume 7, No. 8, August 1993.

[16] Laubach, Mark, "IP Over ATM and the Construction of High-Speed Subnet Backbones," *ConneXions,* Volume 8, No. 7, July 1994.

Documents 1–3 are available from Global Engineering, 15 Inverness Way East, Englewood CO 80112–5704, USA. Phone 1-800-854-7179.

Document 4 is an Internet Draft and represents work in progress; it is available via anonymous FTP from various IETF repositories.

Documents 5–9 are available via anonymous FTP from Internet host `playground.sun.com` in directory `/pub/fibrechannel`. Hardcopies may be obtained from FCSI Administration: contact Gladys Van Polanen Petel, FCSI Administration, Sun Microsystems, M/S UMPK 12–301, 2550 Garcia Ave, Mountain View CA 94043–1100, USA.

More about Fibre Channel:

Special Session: SS4
BOF: Tuesday night

See Program Guide and BOF flyer for details.

PETER WALFORD is an independent industry consultant specializing in high-speed networking and 3-D graphics applications. A 1980 graduate of Amherst College, his experience includes design and implementation of the X.25 and OSI-based link between the internal networks of banks in France and the French inter-banking network installed in 1987, extensive performance analysis and recommended design improvements to the 3-D graphics subsystems in Digital's MIPS-based workstations, and development of a distributed rendering package for clustered workstations. Prior to his experience in France, he directed the implementation of a high-speed fiber optic network supporting distributed geophysical applications for a major oil company. He is presently serving as Technical Lead for the Fibre Channel Systems Initiative (FCSI) Profile development teams and is the editor of the FCSI SCSI and IP Interoperability Profiles. He can be reached via e-mail at: `walford@btr.com`

ED FRYMOYER is currently Program Manager for the Fibre Channel Systems Initiative (FCSI) and Program Manager of the HP/IBM Fibre Channel Alliance. Previously, Frymoyer's responsibilities for Hewlett-Packard included Manager of Long Range Planning and Vision for Communications Components Division and Strategic marketing and business development for new business areas. In these capacities, Frymoyer was in charge of matching HP technical strengths to emerging markets, tracking and projecting for external technical markets and developing division skills and methods to succeed in new markets. His technical expertise includes opto-electronic, computer data networks and telephony (fiber optic and cellular) applications. Frymoyer's educational credentials include Ph.D Physics (Acoustics), Penn State University, MSEE, Northeastern University, BS Eng. Sci., Penn State University. Ed Frymoyer has published a variety of papers and conducts seminars on a regular basis. E-mail: `70523.3010@compuserve.com`

SNMP++:
A Portable Object Oriented Approach to Network Management Programming

by Kim Banker and Peter E. Mellquist, Hewlett-Packard Co.

Introduction

Various *Simple Network Management Protocol* (SNMP) [1] *Application Programming Interfaces* (API) exist which allow for the creation of network management applications. The majority of these APIs provide a large library of functions which require the programmer to be familiar with the inner workings of SNMP and SNMP resource management. In addition, most of these APIs are platform specific, resulting in SNMP code specific to an operating system, network operating system, or network management platform. Application development using **C++** has entered the main stream, and with it a rich sets of reusable class libraries are now readily available. Up to now, a standard set of well designed **C++** classes for network management have been missing. An object oriented approach to SNMP network programming provides many benefits including portability, ease of use, safety, and extensibility. SNMP++ offers power and flexibility which would otherwise be difficult to implement and manage.

What is SNMP++?

SNMP++ is a set of **C++** classes which provide SNMP services to a network management application developer. SNMP++ is not an additional layer or wrapper over existing SNMP code. Rather, SNMP++ coexists with existing SNMP libraries in a few minimized areas and in doing so is efficient and portable. SNMP++ is not meant to replace other existing SNMP APIs or libraries such as WinSNMP, instead it offers power and flexibility which would otherwise be difficult to manage and implement. SNMP++ brings the object oriented advantage to network management programming!

Features

The most important features of SNMP++ are:

- Portability across Operating Systems and Network Operating Systems
- Full SNMP version I Support
- Get, Set and Get-Next
- Trap Reception
- Session Model
- Transport Layer Independence
- Blocked or Asynchronous Modes
- Extensibility through **C++** Sub-classing

SNMP++ offers full functionality for all SNMP version I services. This includes, SNMP *get, set, get-next* and *trap* reception. SNMP++ is designed to be transport layer independent. SNMP++ supports a session model, where a session is defined by a logical connection to a managed agent. Sessions are managed through different instances of SNMP++ SNMP objects. An application may utilize multiple sessions, each corresponding to a different, or same agent. The maximum number of sessions are limited only by the number of available *User Datagram Protocol* (UDP) or *Internet Packet Exchange* (IPX) sockets. SNMP++ supports two modes of operation, *blocked* and *asynchronous*. Blocked mode provides automatic retry and timeout for *Protocol Data Unit* (PDU) delivery. Together, blocked and asynchronous modes, give the application developer the flexibility to develop powerful network management applications.

continued on next page

SNMP++ *(continued)*

SNMP++ example

Perhaps the best way to introduce SNMP++ is through a small example which illustrates its power and simplicity. The following example is taken from a Microsoft Windows application, but is portable across other platforms. The example obtains an SNMP *Management Information Base* (MIB) system descriptor object from the specified agent. Included is all code needed to create an SNMP session, get an octet variable, and print it out. Transmission retry, time-out and SNMP resource allocation are managed automatically. Those readers who have written similar code using WinSNMP or other comparable SNMP APIs should appreciate the simplicity and power which SNMP++ provides.

```
//--------[ SNMP++ get system descriptor example code ]---------
#include "snmp.h"
void get_system_descriptor( HWND hWnd)
{
  long int err_status, err_index;  int status;  char msg[255];
  Vb vb;                                  // Construct a snmp++
                                          // variable binding object
  vb.set_oid("1.3.6.1.2.1.1.0");          // Load the vb object with
                                          // sys descriptor ID

  // Construct an SNMP object with xport type,
  // dest context, & return status
  Snmp snmp( hWnd, (Protocol) ip, "public", &status);

  // Invoke get member function at agent IP address 15.29.33.10
  status = snmp.get( &vb, 1, err_status, err_index,16, "15.29.33.10");
  if ( status == 0)
    vb.get_value( msg);                   // Extract a char array from
                                          // the returned vb object
  else
    sprintf( msg, "SNMP++ Error = %d", status);
                                          // Error from get request

  // Display the sys descriptor ID value
  MessageBox( hWnd, msg, "SNMP++ Demo", MB_OK);

};   // That's all !
```

The actual SNMP code is made up of only five calls. Two SNMP++ objects are utilized, the *Variable Binding* (Vb) and the SNMP object. The Vb object is constructed and two public member functions are utilized. `Vb::set_oid`, sets the Oid portion of the Vb object. `Vb::get_value` extracts an octet array from the returned Vb object. SNMP++ *set* and *get-next* requests are almost identical. That is, they utilize identical parameter lists.

Portability

A major design goal of SNMP++ was to provide a portable SNMP API across a variety of operating systems (OSs), network operating systems (NOSs), and network management platforms. This is accomplished by hiding the internal mechanisms of SNMP and providing a generic interface to SNMP services. A program which uses SNMP++ does not have to change SNMP code to move to another platform.

Platforms which SNMP++ have been successfully deployed include:

- MS-Windows 3.1
- MS-Windows For Work Groups 3.11
- MS-Windows NT (Win16 subsystem)
- MS-Windows '95 Beta II (Win16 subsystem)
- HPUX version 9.0, series 700 & 800 workstations

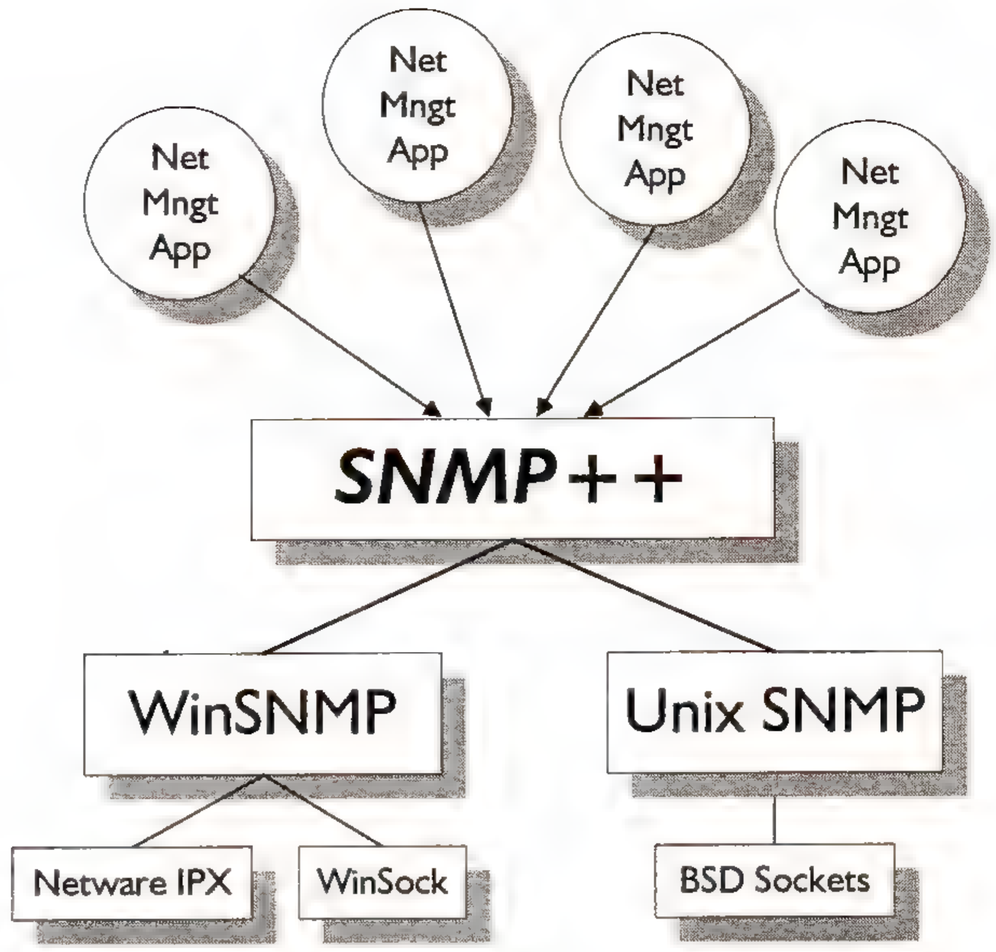


Figure 1: Portability of SNMP++

Another issue in the area of portability is the ability to operate over a variety of transport layer protocols. SNMP++ currently operates over the Internet Protocol (IP/UDP) or IPX protocols, or both using a protocol stack that supports both IP and IPX concurrently.

Usability

An Object Oriented approach to SNMP programming should be easy to use. After all, SNMP is supposed to be a *simple* network management protocol! An application programmer should not need be concerned with low level SNMP mechanisms. An object oriented approach to SNMP encapsulates and hides the internal mechanisms of SNMP, and in doing so provides safety from inadvertently doing the wrong thing. A user does not have to be an expert in SNMP to use SNMP++. Furthermore, a user does not have to be an expert in **C++**! Basic knowledge of SNMP is required, but only a minimal understanding of **C++** is required.

In addition to providing ease-of-use, SNMP++ preserves SNMP flexibility. A number of commercial applications have been successfully created using SNMP++. SNMP++ is designed to be flexible and multipurpose. Much of this flexibility and power would be difficult to implement and manage using a traditional API. To permit greater flexibility and extensibility, the application programmer may subclass the base SNMP++ classes to provide specialized behavior and/or attributes. This theme is central to object orientation. The base classes of SNMP++ are meant to be generic and do not contain any vendor specific data structures or behavior. New attributes can be easily added through **C++** sub-classing and member function redefinition [5].

Most SNMP APIs require the programmer to manage a variety of resources [2]. These include:

- Object Id's (Oids)
- Variable Bindings (Vbs)
- Variable Binding Lists (Vbls)
- Protocol Data Units (PDU)
- Context Names
- Authentication Structures

continued on next page

SNMP++ *(continued)*

Improper allocation or deallocation of these resources often results in corrupted or lost memory. SNMP++ provides safety by managing these resources internally. A user of SNMP++ does not have to be concerned with providing reliability for an unreliable transport mechanism. SNMP++ includes built-in error checking and automatic timeout and retry mechanisms to provide the user with transparent reliability.

SNMP++ classes

SNMP++ is based around three **C++** classes; the Oid Class, the Vb Class, and the SNMP class. SNMP++ was designed using the *Object Modeling Technique* (OMT). OMT is a well known Object Oriented Methodology which allows detailed modeling of objects including their relationships and associations [3].

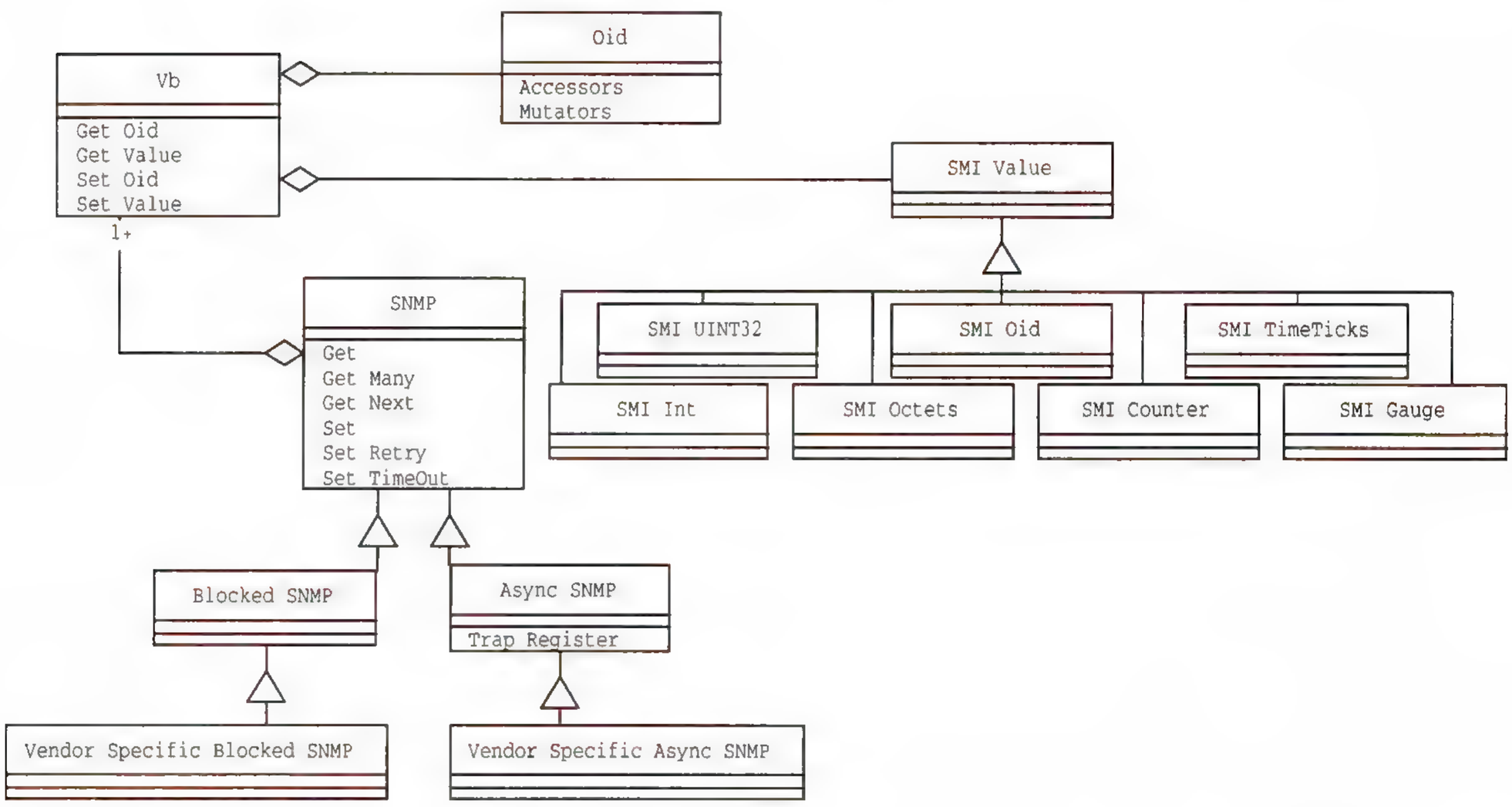


Figure 2: OMT (Object Modeling Technique) View of SNMP++

Oid class

The *Object Identification* (Oid) class is the encapsulation of an SNMP object identifier. The *Structure of Management Information* (SMI) Oid, its related structures and functions, are a natural fit for object orientation. In fact, the Oid class shares many common features to the **C++** String class [7]. For those readers familiar with the **C++** String class or *Microsoft Foundation Classes* (MFCs) Cstring class, the Oid class will be familiar and easy to use.

The Oid class is designed to be efficient and fast. Do not make the false assumption that by using **C++**, a performance penalty must be paid. A well encapsulated **C++** class is easier to fine tune than the equivalent **C** code [4]. The Oid class allows definition and manipulation of object identifiers, mostly through overloaded **C++** operators. The Oid Class is fully portable and does not rely on WinSNMP or any other Windows or Unix SNMP API to be present. The Oid class may be compiled and used with any ANSI **C++** compiler. The Oid class includes all the related SMI types for Oids. A complete description of the Oid class can be found in the SNMP++ document [6].

```
//-----------------[ SNMP++, Oid Class Examples ]-----------------
// A few simple Oid Class examples, see the SNMP++ document
// for all examples
void oid_examples()
{
  Oid o1;                          // construct an empty oid object
  Oid o2( "1.2.3.4.5");            // construct an oid using a dotted string
  Oid o3( o2);                     // construct an oid using another
                                   // oid object

  o3 += 1;                         // concat a 1 to o2
                                   // making it 1.2.3.4.5.1
  printf( "%s",o3.strval());       // print out o3 as a dotted string
  o3 = "1.3.6.8";                  // reassign o3 to a different
                                   // dotted string

  if ( o3 == "1.3.6.8")            // overloaded equivalence operator for
    printf( "o3 = 1.3.6.8");       // comparison

} // automatic cleanup when Oids go out of scope!
```

Vb class

The *Variable binding* (Vb) class represents the encapsulation of an SNMP SMI variable binding. A variable binding is the association of an SNMP object ID with its SMI value. In object oriented methodology, this is simply a "has a" relation. A Vb object "has a" Oid object and "has a" SMI value. The Vb class allows the application programmer to instantiate Vb objects and assign the Oid and value portions. Conversely, the Oid and value portions may be extracted. The public member functions are overloaded to provide the ability to set or get different values to the Vb binding. Variable binding lists in SNMP++ are represented as arrays of Vb objects. All SMI types are provided within the Vb Class. The Vb class provides full data hiding. The user does not need to be concerned about SMI value types, Oid representations, or other related SNMP structures. Like the Oid class, the Vb class is fully portable using a standard ANSI C++ compiler and is not dependent on other SNMP libraries.

```
//-----------------[ SNMP++, Vb Class Examples ]-----------------
// A few simple Vb Class examples, see SNMP++ document
// for all examples
void vb_examples()
{
  Vb vb1;                          // construct a single vb object
  Vb vbs[10];                      // construct an array of
                                   // 10 Vb objects
  Oid oid1("1.2.3.4.5");           // construct an oid object
                                   // with a dotted string
  int x;
  Oid oid2;                        // construct an empty oid object
  vb1.set_oid("1.2.3.4");          // assign the oid portion of the vb
  vbs[0].set_oid( oid1);           // set the oid portion of vbs[0]

  vb1.set_value( (int) 22);        // set value portion of vb1
                                   // to an int, value 22
  vbs[0].set_value( oid1);         // set value portion of vbs[0]
                                   // to an oid, oid1

  vb1.get_value ( x);              // extract an integer value
                                   // from vb1
  vbs[0].get_value( oid2);         // extract oid object from vbs[0]

} // automatic cleanup when Vbs go out of scope!
```

continued on next page

SNMP++ *(continued)*

SNMP class

The most important class in SNMP++ is the *SNMP class*. The SNMP class is an encapsulation of a SNMP session. An SNMP session includes a logical connection from an SNMP management station to a managed agent or agents. Handled by the session are the construction, delivery and reception of SNMP PDUs. Most APIs require the programmer to directly manage the session. The SNMP class manages a large part of the session and frees the implementor to concentrate on the agent management. Through utilization of the SNMP class for session management, the implementor may reuse well developed and tested code. The alternative is to design, implement and test your own SNMP engine. The SNMP class manages a session by:

- Managing the transport layer over a UDP or IPX connection.
- Serializes and un-serializes the SNMP++ objects
- Provides for delivery and reception of PDUs
- Manages all necessary SNMP resources.

The SNMP class is easy to use. Three basic methods, *Snmp::get, Snmp::set* and *Snmp::get_next* provide the basic functions for a network management application. Multiple sessions may be used, each asynchronously firing simultaneous requests.

The SNMP class interface is portable across OSs and NOSs. The implementation of the SNMP class is platform specific. That is, it is implemented for each OS and NOS platform. Currently this module has been ported to run on MS-Windows over WinSNMP and HPUX. The amount of coding needed to port SNMP++ to other platforms is minimal. To the application programmer, no code changes are required to move from one platform to the next. The vast majority of SNMP++ is re-used across platforms; thus, development time and testing time are cut drastically.

Conclusion

SNMP++ is an API created at Hewlett-Packard Roseville Networks Division that permits network management application developers to create SNMP applications. SNMP++ is a very easy to use, portable, flexible, and extensible set of C++ classes that allows a programmer to focus more on the network management application and less on the details and maintenance of a complex SNMP API.

SNMP++ is more than just an API. SNMP++ is also a field tested working implementation. There are a variety of HP network management applications which have been successfully implemented and ported across platforms using SNMP++. Possible future directions for SNMP++ will include more OS and NOS support and enhancements for SNMP version 2.

References

[1] Case, J. D., McCloghrie, K., Rose M. T., & Waldbusser, S. L., "Introduction to version 2 of the Internet-standard Network Management Framework," RFC 1441, April 1993.

[2] Case, J. D., McCloghrie, K., Rose M. T., & Waldbusser, S. L., "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 1442, April 1993.

[3] Rumbaugh, James, *Object-Oriented Modeling and Design,* Prentice Hall, 1991.

[4] Meyers, Steve, *Effective* ***C++****,* Addison Wesley, 1994.

[5] Ellis, Margaret A., Stroustrup, Bjarne, *The Annotated* ***C++*** *Reference Manual,* Addison Wesley, 1990.

[6] Mellquist, Peter E., "SNMP++ An Object Oriented Approach to Network Management Programming Using **C++**," Sunsite anonymous FTP server:
`pub/micro/pc-stuff/ms-windows/winsnmp/snmp_pp.doc`.

[7] Stroustrup, Bjarne, *The* ***C++*** *Programming Language,* Second Edition, Addison Wesley, 1991.

[8] *ConneXions,* Two *Special Issues* on Network Management and Network Security, Vol. 3, No. 3, March 1989 and Vol. 4, No. 8, August 1990.

[9] Case, J. D., Davin, J. R., Fedor, M. S., & Schoffstall, M. L., "Network Management and the Design of SNMP," *ConneXions,* Volume 3, No. 3, March, 1989.

[10] Case, J., McCloghrie, K., Rose, M. T., Waldbusser, S., "The Simple Management Protocol and Framework: Managing the Evolution of SNMP," *ConneXions,* Volume 6, No. 10, October 1992.

[11] Rose, M. T., "Network Management is Simple: You just need the 'Right' Framework." In *Integrated Network Management, II,* Iyengar Krishan and Wolfgang Zimmer, editors, pages 9–25, North Holland, April 1991.

[12] Rose, M. T., *The Simple Book: An Introduction to Internet Management,* Second Edition, Prentice-Hall, ISBN 0-13-177254-6, 1993.

[13] McConnell, J., "DMTF: A Foundation for Systems Management?" *ConneXions,* Volume 8, No. 9, September 1994.

[14] Hall, Martin, "Windows Sockets: Where necessity is the mother of *re*-invention," *ConneXions,* Volume 7, No. 9, September 1993.

[15] Allard, J., "Windows Sockets 2.0: The Next Generation Transport API for Microsoft Windows," *ConneXions,* Volume 8, No. 9, September 1994.

[Ed. This article is adapted from a more detailed SNMP++ document available on the WinSNMP anonymous FTP server located at host `sunsite.unc.edu`, in the directory: `pub/micro/pc-stuff/ms-windows/winsnmp`.]

More about SNMP++:

BOF: Wednesday night

See BOF flyer for details.

KIM BANKER is a software technical scientist at Hewlett-Packard Roseville Networks Division. He received his M.Sc. in Computer Engineering from Carnegie-Mellon University in 1979. After working at Bell Laboratories on a large data networking system, Kim joined Hewlett-Packard to work in the areas of control engineering and video technologies. Kim has served as International Chairman and US ANSI Representative for the ISO Session Network Layer Protocol. Currently Kim is investigating next-generation distributed network management solutions. He can be reached at `banker@hprnd.rose.hp.com`

PETER E MELLQUIST is currently a senior software design engineer for Hewlett-Packard Roseville Networks Division. He holds a B.Sc., and M.Sc. in Computer Science from California State University, Chico. He has been involved in the design and development of networking applications including meteorological information systems and object oriented solutions to networking problems. Peter is an active member of the Association for Computing Machinery's (ACM), Object-Oriented Programming Languages, Systems, Languages and Application (OOPSLA) groups. He can be reached at `mellqust@hprnd.rose.hp.com`

The CU-SeeMe Desktop Videoconferencing Software

by Tim Dorcey, Cornell University

Introduction

CU-SeeMe is a desktop videoconferencing system designed for use on the Internet or other IP networks. It runs on Macintosh and Windows platforms and requires no special equipment for video reception beyond a network connection and gray-scale monitor. Video transmission requires a camera and digitizer, a combined version of which can be purchased for the Macintosh for under $100. CU-SeeMe video is represented on a 16-level gray scale, and is provided in either of 2 resolutions: 320 x 240 pixels (half diameter of NTSC television) or 160 x 120 pixels. At this writing, audio is available in the Macintosh version only, with audio processing adapted from the *Maven* program, written by Charlie Kline at the University of Illinois. When network conditions or equipment deficiencies prohibit reliable audio, ordinary telephone connections can be employed. In addition to basic audio/video services, CU-SeeMe offers crude white board capabilities in the form of a "slide window" that transmits full-size 8-bit gray scale still images and allows for remote pointer control. A plug-in architecture has also been developed to allow 3rd parties to write binary modules which extend the capabilities of CU-SeeMe. Two-party CU-SeeMe conferences can be initiated by one participant connecting directly to the other, whereas larger conferences require that each participant connect to a "CU-SeeMe reflector," a Unix computer running CU-SeeMe reflector software that serves to replicate and re-distribute the packet streams.

The main objective in the development of CU-SeeMe was to produce an inexpensive videoconferencing tool that would be usable today. As well as providing direct benefit to its users, we expected that valuable lessons could be learned about how videoconferencing actually works in practice, how the experience should be organized, what features are necessary to support multi-party conferencing, and so on. While others worked to advance the state of the art in video compression, high-speed networking, and other low-level technologies necessary to support high quality videoconferencing, we hoped to facilitate the accumulation of experience that would provide impetus for those efforts and guide their direction.

Similar efforts have focused on Unix workstations, for which several tools are currently available, including *nv* [1], *ivc* [2] and *vic* [3]. In fact, it was Paul Milazzo's [4] demonstration of such a tool in 1991 that inspired development of CU-SeeMe. However, it is our belief that the value of a communication tool is largely determined by the number of people that can be reached with it. We sought to increase the accessibility of videoconferencing by focusing on low-end, widely available, computing platforms. Currently, CU-SeeMe can be found in places ranging from the grade school to the national laboratories—often with a connection between them. It has appeared in over 40 countries [5] and on every continent—including Antarctica [6]. This article presents a brief overview of two central components of the CU-SeeMe software: *Conference Control* and *Video Encoding*.

Conference control

Each participant in a CU-SeeMe conference periodically transmits a single packet that advertises their interests with respect to all of the other participants. These advertisements are termed "OpenContinue" packets, in recognition of the fact that in a connectionless protocol the information necessary to open a connection is no different than that used to continue it. The OpenContinue packet consists of a standard header that is common to all CU-SeeMe packets, followed by a section of general information about the issuing participant.

Then, for each other participant that the sender is aware of, follows a collection of variables that express the sender's interests with respect to that other participant (e.g., I want their video, I want their audio, I want their slides, etc.). Reflectors examine OpenContinue packets to develop source specific distribution lists for the various media involved in a conference and then forward them to all other participants. Because the protocol requires each participant to process dynamic status information for every other conference participant, it does not scale well to large conferences, say larger than about 30. However, it does provide considerable control, in a robust fashion, over the details of smaller conferences. Furthermore, various possibilities, beyond the scope of this discussion, exist for extending the protocol to larger conferences.

The primary motivation for developing the reflector software was the absence of multicast capabilities on the Macintosh. We have therefore been careful about extending its role beyond the replication and distribution of packets, allowing it to add value where it can, but avoiding dependence on it. We have, however, increasingly come to appreciate the degree to which the reflector architecture allows for fine tuning of the data streams sent to each recipient, and we do not expect this to become any less important when multicast becomes more widely available.

Video encoding

The predominant objective here was to devise algorithms that would be fast enough for real time videoconferencing on the typical Macintosh platforms that were available in mid-1992, which mainly consisted of 68020 and 68030 based machines. The decoding algorithm, in particular, needed to be extremely efficient in order to support multiple incoming video streams. The main technique for achieving these goals was to begin with a massive decimation of the input video signal, and then to process what remained in a manner that took maximal advantage of the capabilities of the target processors, as described below. For simplicity, discussion will focus on the smaller size video format, which has become most popular in practice. Video processing proceeds in 3 basic steps: 1) decimation 2) change detection and 3) spatial compression.

The first step in the video encoding is to decimate the captured 640 x 480 pixel video frame down to 160 x 120, with each pixel represented on a 4-bit gray scale. In comparison to full size, 16-bit color, this represents a 64:1 reduction in the amount of data to be handled by subsequent processing. The user is provided with brightness and contrast controls to adjust the mapping of input intensities to the 16 level gray-scale. With proper adjustment and reasonable lighting conditions, surprisingly good picture quality can be achieved.

Next, the video frame is subdivided into 8 x 8 pixel squares, and a square is selected for transmission if it differs sufficiently from the version of it that was transmitted most recently. The index used to measure square similarity is the sum of the absolute value of all 64 pixel differences, with an additional multiplicative penalty for pixel differences that occur nearby one another. Inclusion of the multiplicative penalty was based on the assumption that changes in adjacent pixels are more visually significant than isolated changes. Its exact form was dictated by computational convenience, devised so as not to introduce any additional computational burden except during the initialization of a look-up table at program load time.

continued on next page

CU-SeeMe *(continued)*

To account for the possibility that updates may be lost in transit, transmission is also triggered if a square has not been transmitted for a specified number of frames (refresh interval). This ensures that a lost update will not corrupt the image into the indefinite future.

Once a square has been selected for transmission, a locally developed lossless compression algorithm is applied. The most interesting feature of this algorithm is the degree of parallelism it achieves by manipulating rows of 8 4-bit pixels as 32-bit words. This allows for high-speed performance on a 32-bit processor, but also complicates exposition of the algorithm. The basic idea is that a square row is often similar to the row above it, and, when it is different, it is likely to be different in a consistent way across columns. Letting *r[i]* represent a 32-bit word containing the *i*th row of pixels in a square, compression is based upon the representation: *r[i] = r[i–1] + d[i],* where *d[i]* is constructed from either a 4, 12, 20 or 36 bit code. If *d[i]* is thought of as being composed of 8 4-bit pixel differences, then spatial redundancy in the vertical direction suggests that the differences will all be near to 0, whereas correlation in the horizontal direction suggests that they will be near in value to each other. Under those assumptions, the sorts of *d[i]* that are most likely to occur can be predicted and a scheme devised to represent them using a relatively small number of bits. Roughly speaking, for each *d[i],* a 4-bit code is used to specify a) a common component of all pixel differences (restricted to the range *[–2, 2]*) and b) whether there are 0 ,8, 16 or 32 bits of additional data to represent deviations around that constant component. In reality, of course, *d[i]* is not composed of individual pixel differences, since carry bits can occur in the 32-bit arithmetic, but the technique still seems to work reasonably well, achieving around 40% compression (compressed size is approximately 60% of original). Although a 40% compression ratio may not appear impressive, recall that this is for images that have already undergone a 64:1 decimation from the original, and that the information discarded at the outset was that most suitable for compression.

The CU-SeeMe video encoding has proven to be surprisingly robust against packet loss when the subject matter is a typical talking head. Often, the only observable effect of high packet loss is a reduction in frame rate. This can be explained as follows. After decimation and compression, it is almost always the case that the information required to update a frame will fit within a single (less than 1,500 byte) packet. Hence a lost packet corresponds to a lost frame update, rather than a partial frame update. Secondly, when the subject is a talking head, most squares are either changing every frame or not at all. Hence a square update that is lost was likely to be replaced in the next frame anyway. Only when a square changes, but then does not change in the next frame, will corruption occur.

This suggests a simple method for embedding several frame rates within a single video stream. Say, every 3rd frame, transmit a square if it is different than it was in the preceding frame *or* if it is different than it was 3 frames ago. Recipients who desire a slower frame rate could accept every 3rd update and still get a clean video image. The observations regarding packet loss suggest that this would not introduce a great deal of additional traffic i.e., a square that differs from 3 frames ago is also likely to differ from the preceding frame. Some variation on this scheme will be implemented in an upcoming version of CU-SeeMe to better support conferences involving participants with differing network capacity.

Conclusions

CU-SeeMe employs a conference control protocol that has proven to be quite robust and allows for the expression of detailed state regarding the relations of each conference participant to each other participant. In conjunction with the reflector software it allows for customized distribution of conference media, so that nothing is transmitted unless it will be used. The protocol is limited in the size of a conference which it can serve, but it can be extended.

CU-SeeMe video is encoded in an ad hoc format that was designed for a particular family of desktop machine that were widespread in mid 1992. What it lacks in mathematical elegance, it makes up for in quickness. As computing power increases, it will eventually become obsolete. Nonetheless, it played an essential role in making CU-SeeMe interesting enough to warrant further work, and it, or its derivatives, will continue to play an important role for some time.

CU-SeeMe can be obtained from:

```
ftp://cu-seeme.cornell.edu/pub/video
```

Notes and references

[1] Frederick, Ron, "Experiences with real-time software video compression," In Proceedings of the Sixth International Workshop on Packet Video, Portland, 1994.

[2] Turletti, Thierry, "The INRIA Videoconferencing System (IVS)," *ConneXions,* Volume 8, No. 10, October 1994.

[3] McCanne, Steve & Jacobson, Van, *vic* man page, Available at `ftp://ftp.ee.lbl.com/conferencing/vic` as of December, 1994.

[4] Milazzo, Paul, Informal demonstration of *dvc* at the December 1991 meeting of the IETF in Santa Fe.

[5] This estimate is based on subscribers to the CU-SeeMe mailing list as of January 1995.

[6] "Tomorrow's TV Today," *Time,* October 10, 1994, p. 24.

[7] Crowcroft, J., "MICE: Multimedia Integrated Communication for Europe," *ConneXions,* Volume 7, No. 11, November 1993.

[8] Sasse, A., Bilting, U., Handley, M., Schulz, C. D., & Turletti, T., "Remote Seminars through Multimedia Conferencing: Experiences from the MICE project," Proceedings of INET '94/JENC5, 1994.

[9] Turletti, Thierry, "A H.261 software codec for videoconferencing over the Internet," INRIA Research Report, No. 1834, Jan. 1993.

[10] Turletti, Thierry, Huitema, C., "Packetization of H.261 video streams," Internet-Draft, September 1994.

[This work was sponsored in part by a grant from the National Science Foundation.]

More about CU-SeeMe:

BOF: Wednesday night

See BOF flyer for details.

TIM DORCEY holds a B.S. from Northern Michigan University, and an M.S. in Psychology and an M.S. in Statistics from Purdue University. Since 1989 he has worked for Cornell Information Technologies, first as a statistical software consultant, and more recently as part of the campus network planning group. Working part-time with Dick Cogger in the summer of 1992, he wrote the original version of CU-SeeMe, and, since the fall of 1993, has devoted full-time effort to CU-SeeMe development. E-mail: `T.Dorcey@cornell.edu`

Announcement and Call for Papers

The IFIP/IEEE International Conference on Distributed Platforms, *ICDP '96,* will be held in Dresden February 27–March 1, 1996.

Background

Client–Server applications are of increasing importance in industry, and have been enhanced with advanced distributed object-oriented techniques, dedicated tool support and both multimedia and mobile computing extensions. Such solutions are a significant step towards a global distributed processing model. Recent responses to this trend are standardized platforms and models including the Distributed Computing Environment (DCE) of the Open Software Foundation (OSF), Open Distributed Processing (ODP) and the Common Object Request Broker Architecture (CORBA) of the Object Management Group (OMG).

Topics

ICDP '96 will be a major forum for distributed systems researchers, network developers, service providers, application designers and end users for discussing the latest research and development results with respect to these platforms. Topics of particular interest include, but are not limited to:

- Experiences with distributed applications and standardized platforms (DCE, CORBA, ODP, ONC+, ANSAware and others)
- Distributed platforms in advanced development/research projects
- Network Services (directory, security, file management etc.)
- Distributed application management
- Objects in distributed environments
- Trading concepts and open markets of distributed services
- Quality of service in distributed applications
- Applications of mobile communication systems

Submissions

The conference is organized into a *Technical Stream* where the latest research results will be presented, an *Industrial Stream* where Industrial developments and trends will be discussed and a *Demonstration Stream* which features an exhibition around distributed processing. Original full papers (max. 15 pages) will belong to the Technical Stream. Furthermore, you are invited to submit extended abstracts focusing on practical work for both the Industrial Stream and the Demonstration Stream (max. 5 pages). Accepted papers will be published in the international conference proceedings by Chapman & Hall. All submissions must be sent online (*PostScript*) to the following e-mail address: `ICDP96@ibc.inf.tu-dresden.de`

Important dates

Deadline for Submission:	June 15, 1995
Notification of Acceptance:	September 15, 1995
Camera ready papers:	October 15, 1995

More information

For further information please contact the program chairs:

Prof. Dr. Alexander Schill
Dresden Univ. of Technology
Dept. of Computer Science
D-01062 Dresden
GERMANY

Prof. Dr. Otto Spaniol/Claudia Popien
RWTH Aachen
Computer Science IV
D-52056 Aachen
GERMANY

URL: `http://www.inf.tu-dresden.de/TU/Informatik/IBDR/lsrn/ICDP96`
FAX: +49 351 4575 335

Internet Domain Survey: January 1995

Number of Hosts, Domains, and Nets:

Date	Hosts	Domains	Replied To Ping*	Network Class A	B	C
Jan 95	4,852,000	71,000	970,000	91	4979	34340
Oct 94	3,864,000	56,000	1,024,000	93	4831	32098
Jul 94	3,212,000	46,000	707,000	89	4493	20628
Apr 94	—N/A—					
Jan 94	2,217,000	30,000	576,000	74	4043	16422
Oct 93	2,056,000	28,000		69	3849	12615
Jul 93	1,776,000	26,000	464,000	67	3728	9972
Apr 93	1,486,000	22,000	421,000	58	3409	6255
Jan 93	1,313,000	21,000		54	3206	4998

[* estimated by pinging 1% of all hosts]

More information

The Internet Domain Survey attempts to discover every host on the Internet by doing a complete search of the Domain Name System. The latest results gathered during late January 1995 are listed above. For more information see RFC 1296. This survey was produced by Mark Lottor of Network Wizards and the data is available on the Internet at `http://www.nw.com/`

Write to *ConneXions*!

We'd love to hear your comments, suggestions and questions about anything you read in *ConneXions*. Our editorial address is given below. Use it for letters to the Editor, questions about back issues etc.:

ConneXions—The Interoperability Report
303 Vintage Park Drive, Suite 201
Foster City, CA 94404–1138
USA
Phone: +1 415-578-6900 or 1-800-INTEROP (Toll-free in the USA)
Fax: +1 415-525-0194
E-mail: `connexions@interop.com`
URL: `http://www.interop.com`

Subscription information

For questions about your subscription please call our customer service hotline: 1-800-575-5717 or +1 502-493-3217 outside the USA. This is the number for our subscription agency, the Cobb Group. Their fax number is +1 502-491-8050. The mailing address for subscription payments is P.O. Box 35840, Louisville, KY 40232–9496.

1995 NetWorld+Interop events

NETWORLD+INTEROP 95

Frankfurt, Germany	*May 29–June 2*
Tokyo, Japan	*July 17–21*
Paris, France	*September 11–15*
Atlanta, Georgia	*September 25–29*

CONNEXIONS
303 Vintage Park Drive
Suite 201
Foster City, CA 94404-1138
Phone: 415-578-6900
FAX: 415-525-0194

ADDRESS CORRECTION
REQUESTED

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

CONNEXIONS

EDITOR and PUBLISHER Ole J. Jacobsen